[illegible]
→ Leiner

2

Packet Radio Temporary Note 296

15 P

ANTIMULTIPATH TECHNIQUES FOR PACKET RADIO, III: THE M'ARY CASE

PREPARED FOR:

Telecommunications Sciences Center

SRI International

Menlo Park, California 94025

Packet Radio Project,

ARPA Order Number 2302

Contract MDA 903-80-C-0222

October 17, 1980

GEORGE L. TURIN
—— engineering consultant ——
763 San Diego Road
Berkeley, CA 94707

Packet Radio Temporary Note 296

ANTIMULTIPATH TECHNIQUES FOR PACKET RADIO, III: THE M'ARY CASE

PREPARED FOR:

Telecommunications Sciences Center

SRI International

Menlo Park, California 94025

Packet Radio Project,

ARPA Order Number 2302

Contract MDA 903-80-C-0222

October 17, 1980

GEORGE L. TURIN
— engineering consultant —
763 San Diego Road
Berkeley, CA 94707

ABSTRACT

In previous PRTN's [1], [2], we have restricted ourselves to consideration of antimultipath techniques for binary signalling. Alternative techniques are available when M'ary signalling is used. We consider the M'ary case here. In particular, we are concerned with the relative performances and complexities of binary and M'ary antimultipath receivers that must operate in urban/suburban multipath.

Our conclusion is that the hoped for simplicity of certain high rate (≈ 1 Mbps) M'ary systems is not achieved because of so-called correlation noise. Thus, binary and (say) 16'ary systems are about equally complex and have more or less equivalent performances. The choice between them must be made on other bases, e.g., interoperability and bandwidth considerations.

TABLE OF CONTENTS

I. Introduction

In previous Packet Radio Technical Notes [1], [2], we have restricted ourselves to consideration of antimultipath techniques for binary signalling. We have seen that, for typical urban multipath spreads (≈5 μs), the attempt to achieve megabit-per-second (Mbps) rates forces us into the complexity of estimator/correlator receivers -- realized as RAKE or DRAKE structures driven by a self-adaptive multipath estimation module. Even with such complex receivers, we have found that multipath-induced intersymbol interference takes a toll of several dB of transmitter power.

An alternative approach is to use M'ary signalling to avoid intersymbol interference. The idea here is to increase the signal duration to the point where it exceeds the multipath spread, while holding the bit rate constant. This, of course, entails using larger-than-binary alphabets. For example, if a rate of 1 Mbps is required, and a multipath spread of 4 μs is assumed, then the signal alphabet must contain at least $2^4 = 16$ signals of duration 4 μs or more.

It is not clear whether using such large signalling alphabets constitutes an advantage. While we might avoid RAKE-like receivers and be able to use simpler post-detection integrating (PDI) receivers, we might also be forced into another sort of complexity -- banks of 16 or more matched filters in all receivers, and the same number of signal generators in all transmitters. It might be a case of "out of the frying pan, into the fire." And, even if the fire is less complex than the frying pan, there is no guarantee that, say, an M'ary PDI receiver will perform as well as a binary RAKE receiver.

On the other hand, in the code-division multiple access (CDMA) context, larger-than-binary alphabets may have advantages unconnected with anti-multipath considerations [3]. Further, in an M'ary system Mbps rates can be achieved with a given processing gain TW by using a factor of $1/\log_2 M$ less bandwidth than in a binary system, a factor that also relieves the timing/sampling accuracies required.

In the present report, we analyze and simulate the performances of several M'ary systems in the multipath environment, and compare them with those of binary systems.

In Section II, we analyze "standard" M'ary systems, in which fixed M'ary alphabets of signals are used. Typically, for urban/suburban multipath having spreads of 4 μs, and for signalling rates of about 1 Mbps, we need 16'ary or larger alphabets to avoid multipath-induced interference with such a system.

In Section III, we consider so-called M'ary commutation codes, in which the alphabet changes with time. The idea of these codes is to reduce the size of the signal set and the number of associated matched filters by deleting from the signal set all recently used signals for as long as it takes for their multipath responses to die out. A typical commutation code for urban multipath using a 1 Mbps rate, suggested by Leung [3], would be 5'ary. Each signal would last 2 μs; only four signals would remain in the code at any time, the most recently used signal being excluded.

The analyses of Sections II and III are much oversimplified. We therefore, in Section IV, give the results of realistic simulation experiments conducted by M. A. Kamil, in which M'ary systems with standard and commutation codes are evaluated.

The results of this work can be summarized as follows:

- Despite the promise given by the theoretical analysis that M'ary PDI (say M = 16) will be be an effective high-rate antimultipath technique involving much less system complexity than binary RAKE/DRAKE, simulation shows that it performs quite poorly. The discrepancy between theory and simulation is due to an oversimplification in the theory, which ignores crosscorrelation noise. This type of noise becomes a predominant factor at large SNR, and greatly limits M'ary PDI performance.

- Standard 16'ary RAKE/DRAKE operates about as well as binary DPSK RAKE/DRAKE, although with much more system complexity.

- 5'ary commutation codes perform neither as well as binary DPSK nor even as well as standard 16'ary codes, whether PDI or RAKE/DRAKE receivers are used.

The fundamental conclusion we come to is that we can do no better in urban/suburban multipath at high (1 Mbps) rates than by using the self-adaptive estimator/correlator structure of [2] that uses binary DPSK signalling and RAKE reception. The choice between, say, 16'ary "orthogonally" keyed RAKE and binary DPSK RAKE lies simply in the following issues:

- Is the estimator/correlator of [2] for binary signalling (which must unscramble intersymbol interference) so much more complicated than the estimator/correlators for 16'ary signalling (which have no intersymbol interference to consider) ~~as~~ to outweigh the natural simplicity of a binary transmitter and receiver? [illegible]

- Is the 4-to-1 reduction in bandwidth of 16'ary compared to binary (for a given TW) or the 4-to-1 increase in TW (for a given bandwidth) of substantial importance?
- What are the implications of cross-correlation noise for 16'ary and binary systems in the context of CDMA or, equivalently, of interoperating systems?*

The answers to these questions are not entirely clear. Our intuitive guess is that in most situations the binary system would be preferable.

In what follows, we assume that the reader is familiar with [1] and [2].

As was the case with other PRTN's in this series, the work herein is heavily based on simulation programs developed under a sequence of grants from the National Science Foundation to the University of California (Berkeley), the most recent being Grant ENG 21512; a debt of gratitude is due NSF for its continuing support of the work at UCB. The author is also grateful to his doctoral student, M. A. Kamil, for having programmed the calculations leading to Figures 2-6 and the simulations leading to Figures 11-24; Kamil's work was supported by SRI under the ARPA Packet Radio Project.

*Systems using the same frequency band but different spread-spectrum codes.

II. Approximate Performance Analysis: Standard M'ary Systems

2.1. Transmitter/Receiver Structure

We assume an M-ary transmitter that transmits one of M signal waveforms every T seconds. The waveforms are assumed to be unformly quasi-orthogonal. That is, if we let $s_i(t)$, $i = 1,\ldots,M$, $0 \le t \le T$, be the waveforms, and suppose that they have equal energies, i.e.,

$$\int_0^T s_i^2(t)dt = \mathcal{E} \ , \quad \forall i, \tag{1}$$

then we assume that

$$\phi_{ij}(\tau) \triangleq \int_0^T s_i(t)\ s_j(t-\tau)dt << \mathcal{E} \tag{2}$$

for all $i \neq j$ and all τ. Further, we assume that each waveform has a "good" autocorrelation function, i.e.,

$$\int_0^T s_i(t)\ s_i(t-\tau)dt << \mathcal{E} \tag{3}$$

for all i and for all $\tau > \frac{1}{W}$, where W is the signal bandwidth.

The receiver is specified to consist of M matched filters, one matched to each of the signals, followed by envelope detectors and decision circuitry. The outputs of the envelope detectors, after the isolated transmission* of the kth signal in the interval (0,T), look like the waveforms in Figure 1.

Figure 1 is drawn for the case of large output signal-to-noise ratio. The waveforms depicted for the outputs of all but the kth filter consist of the cross-correlation functions of the form (2), with $k \neq j$; there is one such function present for each of the paths. In addition, the waveforms of Figure 1 have a random noise

The kth filter output consists of correlation peaks in $(T,T+\Delta)$, one for each path, plus "tails" of the kth autocorrelation function (i.e., (2) with $i = j = k$), plus random noise. Note that if another waveform

*I.e., no transmission precedes or succeeds this transmission.

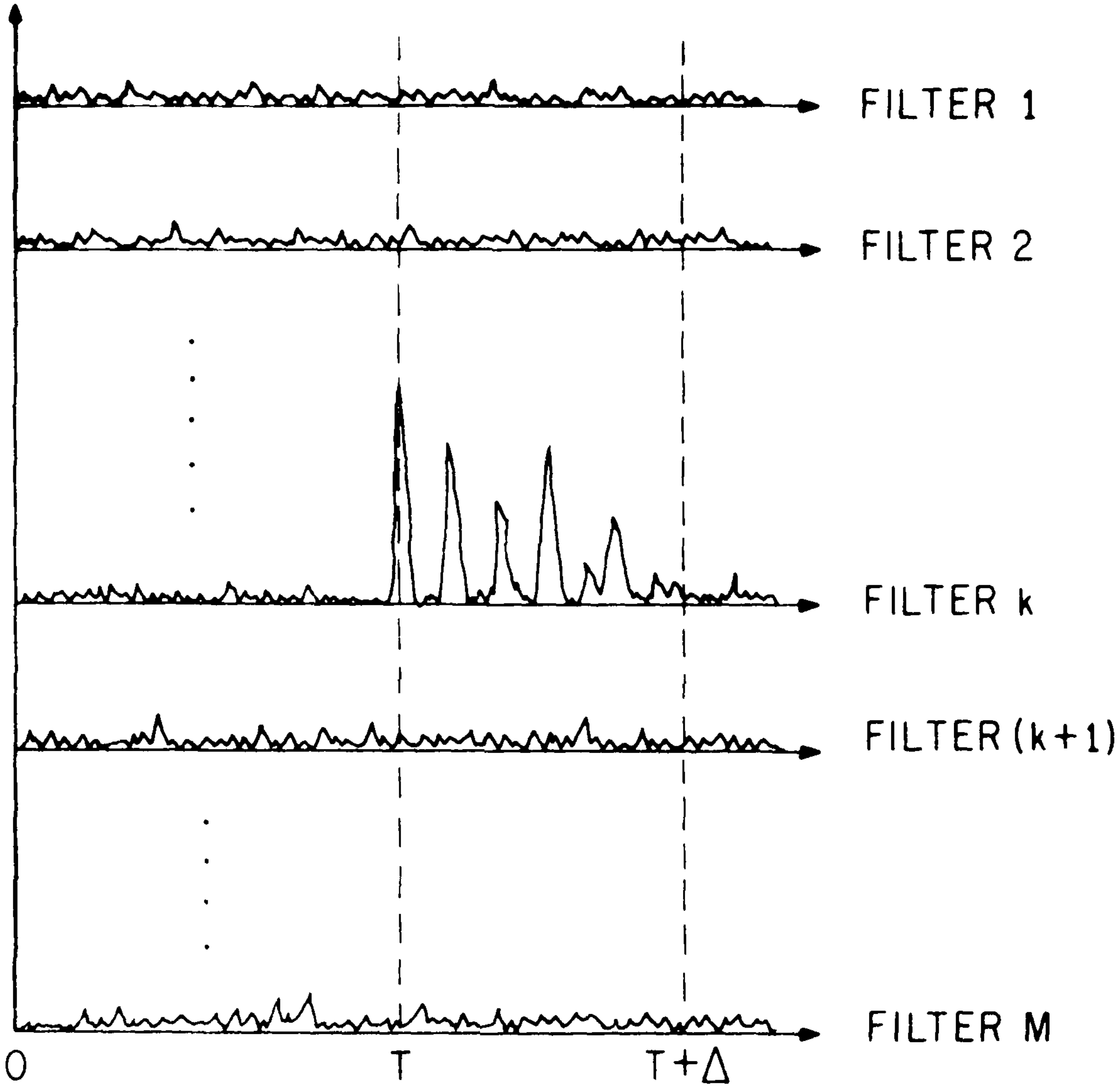


FIGURE 1

had been transmitted in the preceding interval (-T,0), multipath peaks would appear in $(0,\Delta)$ at the output of the corresponding matched filter. Thus, when $\Delta < T$ as shown, even if the same signal should be transmitted twice in a row, the correlation peaks induced in the associated filter output by the two successive transmissions would not be interleaved as, e.g., in Figure 18 of [1]. That is, intersymbol interference will be small and -- as discussed in [1] -- a PDI receiver seems viable.

The decision circuitry can consist of any of the types discussed in previous reports [1], [2]: a "first path" (FP) receiver, which locks onto the first above-threshold path; a "largest path" (LP) receiver, which locks onto the strongest path; a post-detection integrator (PDI), which integrates each detected filter output from T to $T+\Delta$; a weighted post-detection integrator (WPDI); RAKE, which estimates path delays and strengths, samples the filter outputs at the estimated path delays, and combines the samples with appropriate strength-related weights; or DRAKE, which operates like RAKE, but combines all samples with equal weights. In all cases, identical operations are performed on all M filter outputs, and the operation leading to the largest result causes a decision in favor of the corresponding signal.

2.2. Error-Probability Analysis

Ideally, we should derive error-probability expressions for all of the decision mechanisms listed above, and then compare them with each other and with corresponding results for the binary case, given in previous reports. Unfortunately, M'ary error-probability analysis is not a simple matter, so we shall concentrate on approximate results that we can get quickly from well-known starting points. In particular, we shall limit ourselves to the analysis of the FP, PDI and DRAKE receivers.

2.2.1. FP Receiver

The error probability for a non-coherent M'ary system operating through a single, nonfading path is well known to be (see [4],[5]):

$$P_{E,1}^{(M)}(\mathcal{E}) = \frac{1}{M}\sum_{k=2}^{M}(-1)^k\binom{M}{k}e^{-\left(\frac{k-1}{k}\right)\frac{\mathcal{E}}{N_0}} \qquad (4)$$

where $\mathcal{E}$ is the energy given by (1) and N_0 is the white noise power density in the channel. The notation $P_{E,1}^{(M)}(\mathcal{E})$ indicates that this error probability is for the one-path channel and for the full M'ary ($\log_2 M$-bit) character (we discuss its relationship to the bit error rate below), and that it is a function of $\mathcal{E}$, which we subsequently make a random variable. Note that for M = 2 we have the very familiar

$$P_{E,1}^{(2)}(\mathcal{E}) = \frac{1}{2} e^{-\mathcal{E}/2N_0} \tag{5}$$

In order to apply (4) to the FP receiver, we make three assumptions:

- The receiver always locks onto the first "strong" (or above-threshold) path, which is usually the LOS path. (This is not as satisfactory as locking onto the strongest path.) We assume that the identity of this path does not change with time.
- Path capture is perfect, i.e., the receiver samples the path at the precise instant of the signal peak.
- The (auto- or cross-) correlation tails, due to other paths, at the captured time instant are negligible in all M matched filter outputs. That is, at the sampling instant, the multipath-induced interference in all filter outputs is dominated by the channel noise.

Under these assumptions, the FP receiver performs exactly as if there were only one path. If this path is Rayleigh fading, then the energy received by it has an exponential distribution:

$$p_1(\mathcal{E}) = \frac{1}{\bar{\mathcal{E}}} e^{-\mathcal{E}/\bar{\mathcal{E}}}, \quad \mathcal{E} \geq 0 \tag{6}$$

where $\bar{\mathcal{E}}$ is the mean energy. The probability of error for the time-capture receiver is obtained by averaging (4) over (6):

$$\bar{P}_{E,1}^{(M)} = \frac{1}{M} \sum_{k=2}^{M} (-1)^k \binom{M}{k} \frac{1}{1 + \frac{k-1}{k} \frac{\bar{\mathcal{E}}}{N_0}} \tag{7}$$

Again, for M = 2, we obtain the familiar result

$$\bar{P}_{E,1}^{(2)} = \frac{1}{2 + \frac{\bar{\mathcal{E}}}{N_0}} \quad . \tag{8}$$

2.2.2. DRAKE Receiver

We idealize the DRAKE receiver by assuming that:

- The delays of all paths have been estimated with perfect accuracy.
- At each path sampling point the (auto- and cross-) correlation tails from all other paths are negligible in every filter's output. (See further discussion, below.)

Supposing that all paths are Rayleigh distributed with equal mean-square strengths, the outputs of all DRAKE combiners are χ^2 variables with 2K degrees of freedom, where K is the number of paths present (hence, the number of DRAKE taps turned on). The combiner output corresponding to the symbol being sent has energy due to both signal and noise; all other combiners have output energies due (approximately) to noise alone. It is easily shown that the probability of a character error is given by the expression

$$1 - \bar{P}_{E,K}^{(M)} = \int_0^\infty \frac{y^{K-1}}{(K-1)!} e^{-y} \left[\int_0^{(1+\frac{\bar{\mathcal{E}}}{N_0})y} \frac{z^{K-1}}{(K-1)!} e^{-z} dz \right]^{M-1} dy \,. \tag{9}$$

Hahn [4] has shown $\bar{P}_{E,K}^{(M)}$ to be

$$\bar{P}_{E,K}^{(M)} = \frac{(M-1)!}{[1+\frac{\bar{\mathcal{E}}}{N_0}]^K (K-1)!} \sum_{r=1}^{M-1} \frac{(-1)^{r-1}}{(M-r-1)!\,r!} \sum_{j=0}^{r(K-1)} \frac{(K+j-1)!}{[r+\frac{1}{1+\frac{\bar{\mathcal{E}}}{N_0}}]^{K+j}} C_{r,j} \tag{10}$$

where

$$C_{r,j} = \sum_{x=j-(K-1)}^{j} C_{r-1,j} \frac{1}{(j-x)!} \epsilon_x^{0\le x\le (r-1)(K-1)} \tag{11}$$

with

$$\epsilon_x^{a\le x\le b} = \begin{cases} 1 & \text{if } a \le x \le b \\ 0 & \text{elsewhere} \end{cases} \tag{12}$$

and

$$\begin{aligned} C_{0,0} &= 1 \\ C_{1,j} &= 1/j! \\ C_{r,1} &= r \\ C_{r,0} &= 1 \end{aligned} \qquad (13)$$

As expected, (1) reduces to (7) for K = 1.

2.2.3. Post-Detection Integrating (PDI) Receiver

In [1] and [2], theoretical and simulational performance curves for PDI and DRAKE were given for binary, DPSK signalling in the low-rate (no intersymbol interference) case. As expected, PDI performed worse than DRAKE, since PDI is equivalent to a DRAKE delay line with all taps -- even those with only noise present -- always turned on. The margin between PDI and DRAKE was shown to be between 2 and 5 dB, depending on the characteristics of the multipath (geographical area) and whether theoretical or simulational curves were compared.

In estimating the performance of PDI for $M > 2$, we assume that its loss compared to DRAKE is the same for all M, since "turning on the spurious taps" adds the same amount of noise to each of the M outputs, regardless of the value of M.* That is, it would seem that going from DRAKE to PDI effectively increases the system noise level by a given amount, regardless of the value of M. We therefore estimate that, in theory, the performance of M'ary PDI should be 2-5 dB worse than M'ary DRAKE.

2.3. Comparison of Systems

There are at least two ways to compare the performances of systems with different values of M. Both have validity under particular circumstances.

In one case -- which would apply in PCM systems and in packet systems not having error correction -- the probability of a word error

*Of course, the additional noise changes the distributions of the delay-line outputs, thus affecting the error probabilities differently for different M. We ignore this effect, which should vanish for large Δ.

is important. In this case, we rank systems by comparing the energies they require per bit in order to achieve equal probabilities of error for an N-bit word. Thus, for example, if N = 7 and we were comparing M'ary systems with M = 2, 16 and 64 (1, 4 and 6 bits per character), we would compare the energies per bit they require when

$$(1-P_2)^{64} = (1-P_{16})^8 = (1-P_{64})^2 \tag{14}$$

where P_M is the probability of a character error in the M'ary system. Notice that if decoding of M'ary characters into binary sequences is performed by the systems, we are not concerned with the number of binary errors in an N-bit decoded word; a single error in N bits is adjudged as severe as N errors, for each would cause a word error.

An alternative method of comparison is more important for systems with error correction. In such systems, a few isolated bit errors may be correctable, so we cannot equate all patterns of errors in the decoded binary sequences. What we need here is an evaluation of the probability that an individual bit in the decoded sequence is in error. Even if such an evaluation were always easy to do, comparison of systems in this case might still be a problem, since there is no guarantee that the decoded bit sequence for each system has independent bit errors.

We assume for the second method of comparison that the M'ary signalling schemes used are such that all M'ary character errors are equally proable, i.e., that any character in error is equally likely to be any character but the correct one. In that case, we can use a formula of Viterbi [6], which says that individual bit errors in the decoded bit stream are independent and have probability

$$P_{EB} = \frac{M}{2(M-1)} P_E^{(M)} \tag{15}$$

where $P_E^{(M)}$ is the M'ary character-error probability. Note that, for large M, $P_{EB} \cong \frac{1}{2} P_E^{(M)}$. This says that when an M'ary character is received incorrectly, about one-half of the $(\log_2 M)$ bits in the binary sequence into which it is decoded will be in error.

It is not always clear which of tne two methods of comparison is preferable. Fortunately, for the anti-multipath under consideration here, the difference between the two methods is not great, amounting to 1 dB or less of transmitter power.

2.4. Results

The character error probability for the FP receiver ($\bar{P}_{E,1}^{(M)}$ of (7) or (10)) and for DRAKE ($\bar{P}_{E,K}^{(M)}$ of (10)) is plotted in Figures 2-6,* as a function of $\rho = \bar{\mathcal{E}}_b/N_0$, where $\bar{\mathcal{E}}_b = \bar{\mathcal{E}}/\log_2 M$ is the average received energy per bit ($\bar{\mathcal{E}}$ is the average received energy per character). Each figure is for a different value of M, with K as a parameter. We shall use these curves later to construct theoretical plots of the performance of M'ary systems in urban/suburban multipath. However, we shall convert character error probability to bit error probability using (15).

*Programming done by M. Kamil.

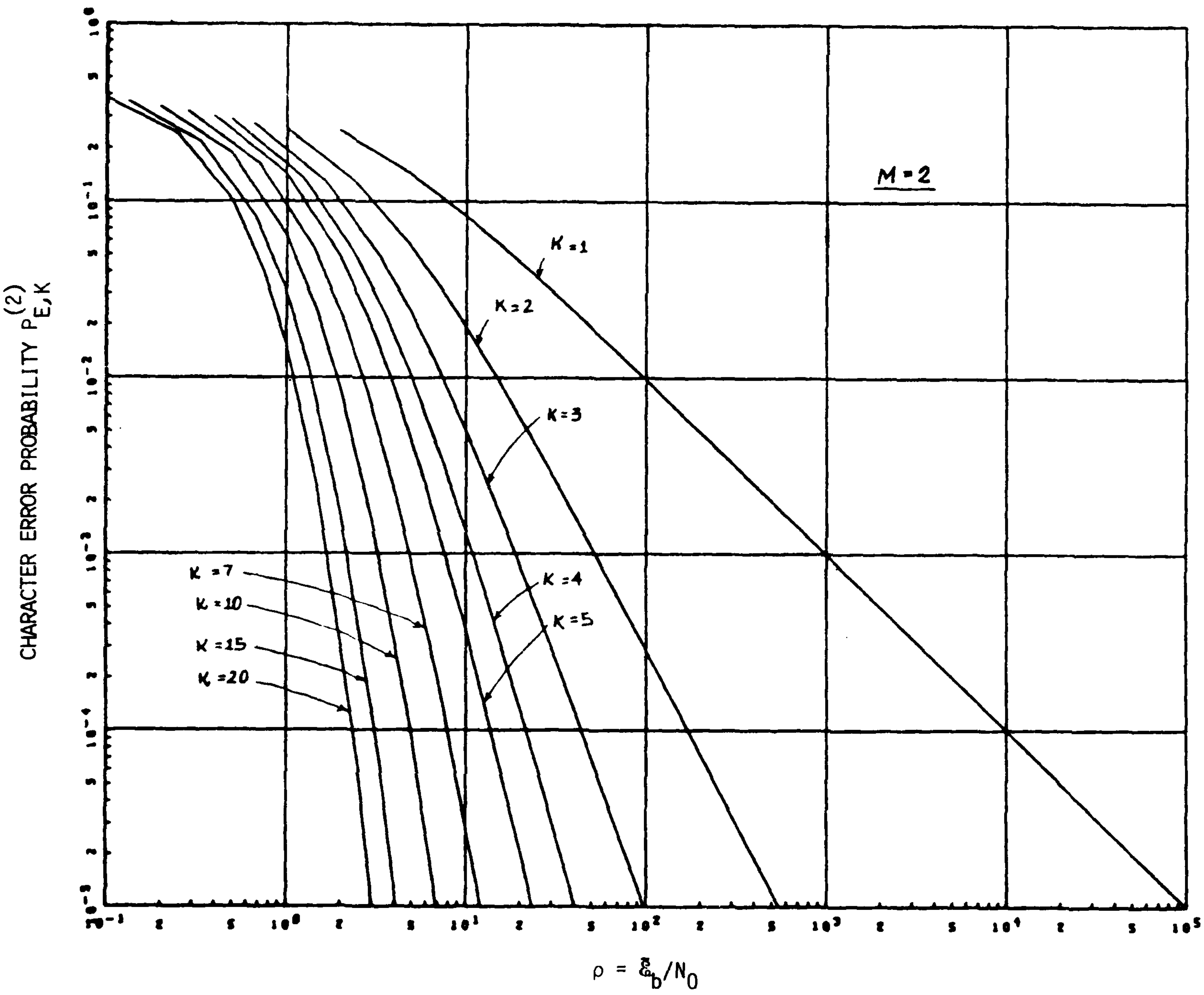

CHARACTER ERROR PROBABILITY $P_{E,K}^{(2)}$
M = 2
K = 1
K = 2
K = 3
K = 4
K = 5
K = 7
K = 10
K = 15
K = 20
10^0
5
2
10^{-1}
5
2
10^{-2}
5
2
10^{-3}
5
2
10^{-4}
5
2
10^{-5}
10^{-1}
2
5
10^0
2
5
10^1
2
5
10^2
2
5
10^3
2
5
10^4
2
5
10^5
$\rho = \bar{\mathcal{E}}_b/N_0$

FIGURE 2

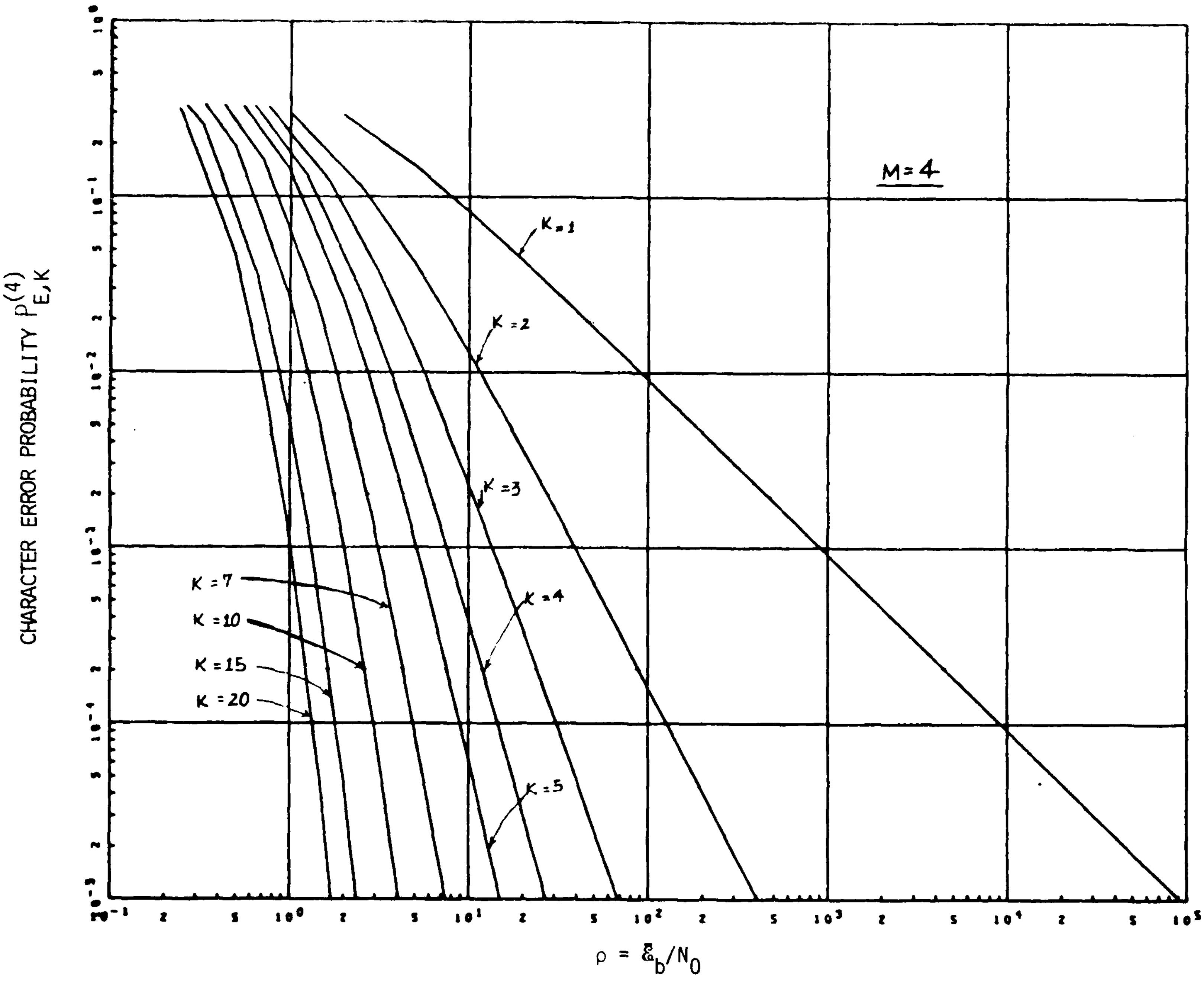

M=4
K=1
K=2
K=3
K=4
K=5
K=7
K=10
K=15
K=20
CHARACTER ERROR PROBABILITY $P_{E,K}^{(4)}$
$\rho = \bar{\mathcal{E}}_b/N_0$
10^{0}
5
2
10^{-1}
5
2
10^{-2}
5
2
10^{-3}
5
2
10^{-4}
5
2
10^{-5}
10^{-1}
2
5
10^{0}
2
5
10^{1}
2
5
10^{2}
2
5
10^{3}
2
5
10^{4}
2
5
10^{5}

FIGURE 3

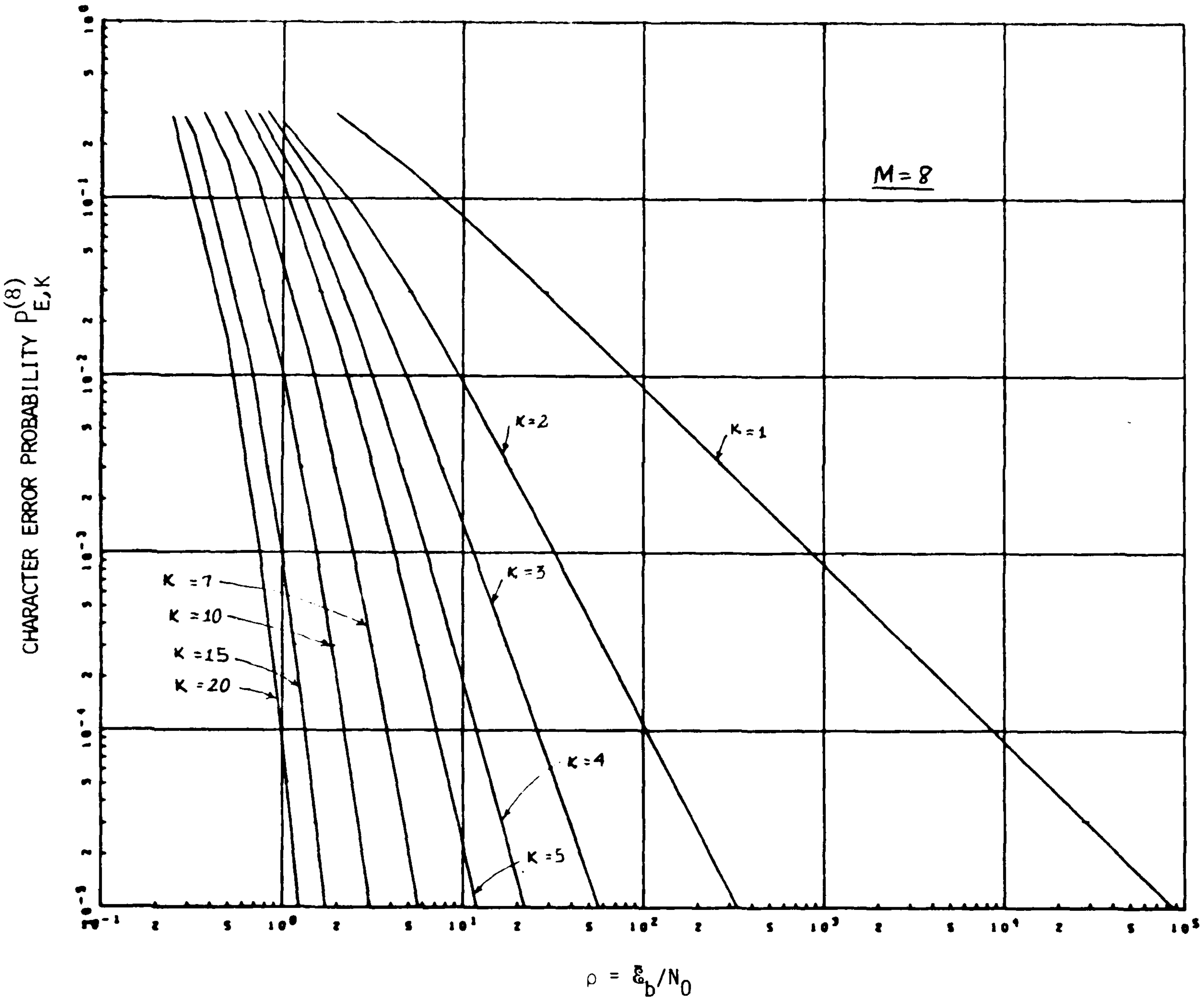

M = 8
CHARACTER ERROR PROBABILITY $P_{E,K}^{(8)}$
$\rho = \bar{\mathcal{E}}_b/N_0$
K = 1
K = 2
K = 3
K = 4
K = 5
K = 7
K = 10
K = 15
K = 20
10^0
5
2
10^{-1}
5
2
10^{-2}
5
2
10^{-3}
5
2
10^{-4}
5
2
10^{-5}
10^{-1}
2
5
10^0
2
5
10^1
2
5
10^2
2
5
10^3
2
5
10^4
2
5
10^5

FIGURE 4

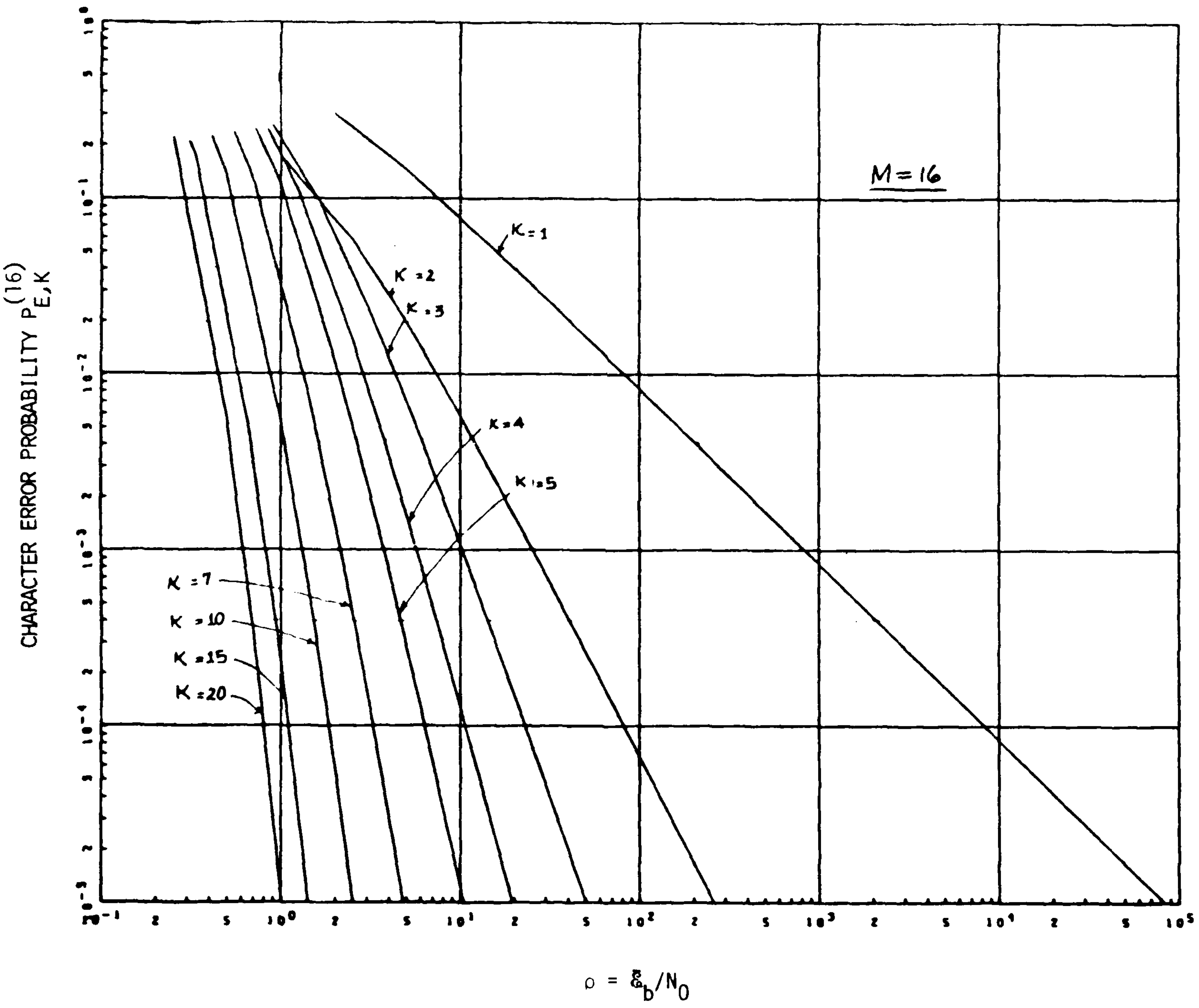

M=16
K=1
K=2
K=3
K=4
K=5
K=7
K=10
K=15
K=20
CHARACTER ERROR PROBABILITY $P_{E,K}^{(16)}$
10^0
5
2
10^{-1}
5
2
10^{-2}
5
2
10^{-3}
5
2
10^{-4}
5
2
10^{-5}
10^{-1}
2
5
10^0
2
5
10^1
2
5
10^2
2
5
10^3
2
5
10^4
2
5
10^5
$\rho = \bar{\mathcal{E}}_b/N_0$

FIGURE 5

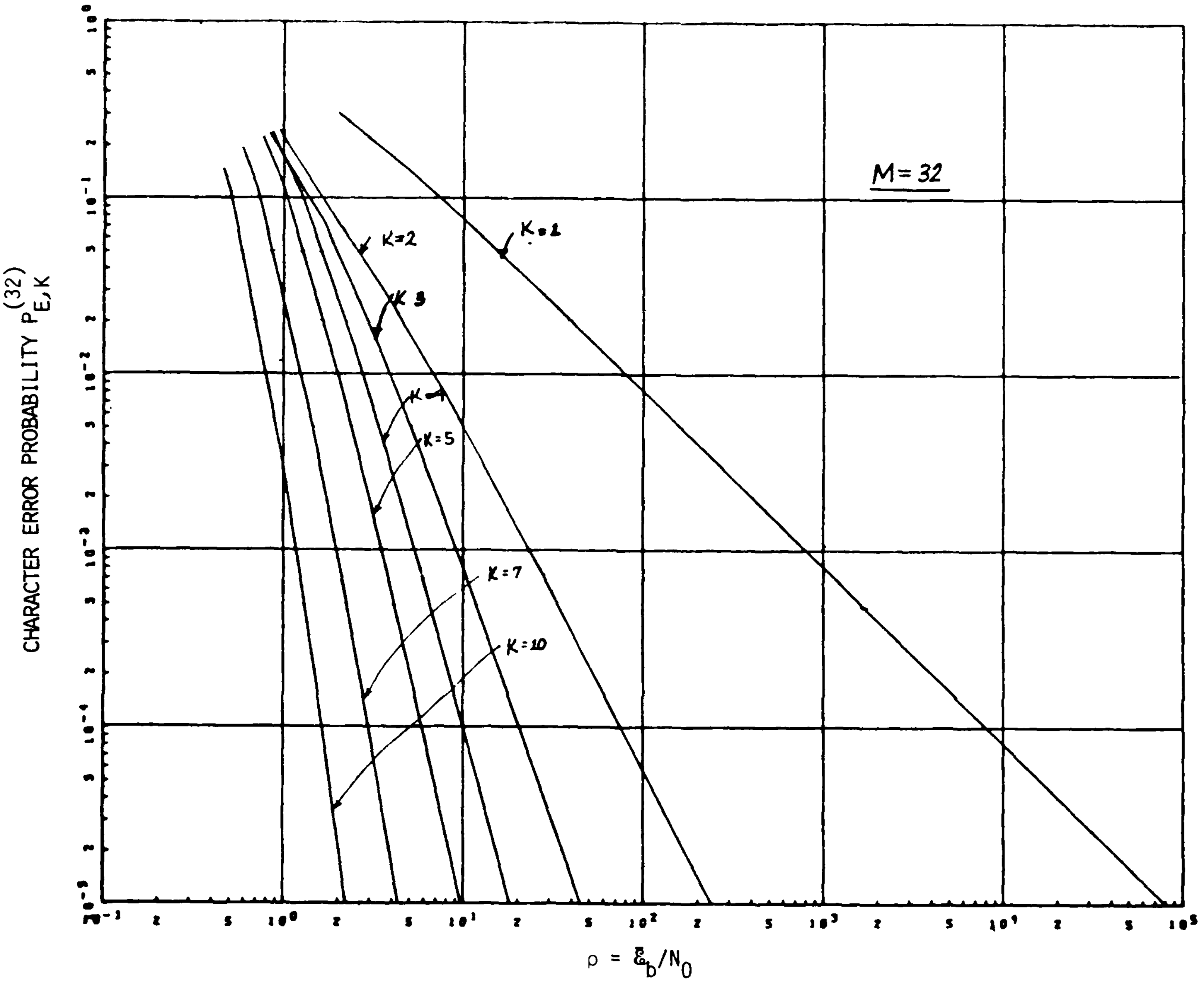

M= 32
K=1
K=2
K 3
K=4
K=5
K=7
K=10
CHARACTER ERROR PROBABILITY $P_{E,K}^{(32)}$
10^{0}
5
2
10^{-1}
5
2
10^{-2}
5
2
10^{-3}
5
2
10^{-4}
5
2
10^{-5}
10^{-1}
2
5
10^{0}
2
5
10^{1}
2
5
10^{2}
2
5
10^{3}
2
5
10^{4}
2
5
10^{5}
$\rho = \bar{\mathcal{E}}_b/N_0$

FIGURE 6

III. M'ary Commutation Codes

3.1. The Concept of Commutation Codes

We analyze here two types of commutation codes:

- Commutation among N M'ary signal sets (NxM codes).
- Commutation among subsets of an (M+N)'ary code ((M+N) or Leung codes).

The purpose of commutation is to delete from use any recently transmitted signal or signal set which is or might be still causing a multipath-channel response. This should allow the use of a PDI receiver.

For example, consider a 4x2 PSK code designed for 1-Mbps operation through a channel with a 4-μs multipath spread. The system would commute cyclicly through the four signal pairs $\pm S_1$, $\pm S_2$, $\pm S_3$, $\pm S_4$ and the outputs of the four associated matched filters.* If S_1, S_2, S_3, S_4 are uniformly mutually quasi-orthogonal in the sense of (3) and each is of duration 1 μs, then the multipath ringing at the output of filter #1 in response to $\pm S_1$ would be over before S_1 was used again, and the same would be true of filters #2, 3 and 4. Thus, we would avoid the interleaving of multipath responses to successive symbols that is characteristic of a high-rate binary system using a single signal pair $\pm S$ (see Figure 18 of [1]), and we should be able to use PDI.

Again, consider a (4+1) code as proposed originally by Leung. Here, one commutes among five mutually uniformly quasi-orthogonal signals, each of 2-μs duration. On each transmission, the transmitter encodes two source bits into a signal chosen from only four of the five signals, excluding from the signal set that signal used just previously. At the receiver, there is a bank of five matched filters, but none can respond twice in a row to its own signal. Hence, each filter can allow a full 4 μs for its multipath response to die out before it is hit again, thus avoiding intersymbol interference and allowing use of PDI. As we shall note, one has a choice here in receiver design: to observe the outputs of M filters (not including that corresponding the last <u>detected</u> signal, which might not be the same as the last transmitted signal), or to observe the outputs of all M+N filters on each reception. We shall discuss

*Alternatively, each binary signal set could be an orthogonal rather than an antipodal pair, but this would require twice the number of matched filters.

the relative merits of these two approaches.

M+N and MxN codes have the following relative advantages and disadvantages:

- MxN codes generally require larger signal sets (hence larger numbers of matched filters) than M+N codes; see Section 3.2.
- Synchronization of signal sets at the transmitter and receiver is more difficult for M+N systems than for MxN systems. For MxN systems, the sequence of signal sets is fixed, and synchronization can be achieved through a separate synchronizing signal or by observing the outputs of the matched filters to establish the current phase of the sequence. In M+N systems, the sequence of signal sets depends on the very data sequence being transmitted. One must make N correct decisions in a row to know correctly which set of M signals will be used next. Otherwise, errors can propagate. See Section 3.3.
- Because of the error propagation feature just noted, M+N codes tend to perform more poorly than MxN codes.

In the remainder of this Section, we explore the characteristics of commutation codes.

3.2. Minimization of the Total Number of Signals Required

As indicated in [1] (see, e.g., Fig. 18 of that report), if a signal's duration is T and the multipath spread is Δ, intersymbol interference due to multipath is completely avoided if no signal is repeated less than Δ+T seconds after its most recent transmission. In practice, this minimum allowable interval can be reduced to Δ seconds, since only low-level auto-correlation sidelobe interference between successive transmissions of a signal will then occur (see Fig. 18 of [1]). We note that whether Δ or Δ+T seconds is allowed, cross-correlation noise due to the response of filters to signals to which they are not matched will be almost equally severe. Thus, allowing Δ seconds between repetitions of a signal is adequate; we henceforth use this spacing.

Let $L = \lceil \frac{\Delta}{T} \rceil$, where $\lceil x \rceil$ is the smallest integer greater than or equal to x. In the context of the commutation codes we have been discussing, we see that intersymbol interference in a matched filter's output can be virtually avoided (except for the low-level autocorrelation sidelobe noise) if repetition of a signal is prohibited for L-1 succeeding transmissions. For example, if T = 1 μs and Δ = 4 μs, no signal should be repeated for $[\frac{\Delta}{T}] - 1 = 3$ transmissions after it has been used.

Let each transmission consist of a choice from an M-ary signal set. MxN codes require N = L signal sets, for a total of MN = ML signals. M+N codes require a single signal set with M+N = M+(L-1) signals, respectively. The question arises: is there an optimal value of M that minimizes the total number of signals required?

Let R be the data rate in bps. Then the signal duration is $T = \frac{\log_2 M}{R}$ and $L = \left\lceil \frac{R\Delta}{\log_2 M} \right\rceil$. Thus, we must minimize

$$Q_{MxN}(M) = ML = M\left\lceil \frac{R\Delta}{\log_2 M} \right\rceil \tag{16}$$

for MxN codes, and

$$Q_{M+N}(M) = M + L - 1 = M + \left\lceil \frac{R\Delta}{\log_2 M} \right\rceil - 1 \tag{17}$$

for M+N codes.

The minimization problem can be solved by letting Q_{MxN} and Q_{M+N} be replaced by continuous functions of M:

$$\tilde{Q}_{MxN}(M) = R\Delta \frac{M}{\log_2 M} \tag{18}$$

$$\tilde{Q}_{M+N}(M) = M + \frac{R\Delta}{\log_2 M} - 1 \tag{19}$$

We then find the minima of each of these functions with respect to M, investigate the two integer values of M that bracket the minimizing M, and use the best of these in (16) and (17).

The minimizing M's for (18) and (19) satisfy

$$\frac{d\tilde{Q}_{MxN}}{dM} = \frac{R\Delta}{\log_2 M}\left[1 - \frac{\log_2 e}{\log_2 M}\right] = 0 \tag{20}$$

and

$$\frac{d\tilde{Q}_{M+N}}{dM} = 1 - \frac{R\Delta \log_2 e}{(\log_2 M)^2 M} = 0 \tag{21}$$

respectively. Of these equations, (9) has an explicit solution, viz., $M_{opt} = e = 2.718\ldots$, for which $\tilde{Q}_{MxN}(M_{opt}) = 1.884\ R\Delta$. For the neighboring integer values M = 2 and M = 3, $\tilde{Q}_{MxN}$ takes on the values $\tilde{Q}_{MxN}(2) = 2\ R\Delta$ and $\tilde{Q}_{MxN}(3) = 1.893\ R\Delta$, respectively, which are plotted as broken-line curves in Figure 7. The minimum number of signals required for MxN codes are, correspondingly,

$$Q_{MxN}(2) = 2\lceil R\Delta \rceil \qquad (22)$$

and

$$Q_{MxN}(3) = 3\lceil 0.631\ R\Delta \rceil \qquad (23)$$

In Figure 7, we have shown for each integer R the preferable of (22) and (23), i.e., the smaller of the two. In case of a tie, the preferable code is the one that has the larger recycle time $NT = \left\lceil \frac{R\Delta}{\log_2 M} \right\rceil \frac{\log_2 M}{R}$; for example, when $R\Delta = 3$, (22) and (23) are both equal to 6, but NT = 3/R for M = 2 and NT = 3.17/R for M = 3, so the (3x2) code is judged better than the (2x3) code.

Note that the optimal MxN oscillates between binary and ternary sets as $R\Delta$ varies. In practice, binary sets are more desirable, and we would normally operate with one, even if slightly non-optimum. Thus, in Figure 7, we would normally use the 2x4 and 2x7 codes rather than the 3x2 and 3x4 codes, respectively.

Solution of (21) involves a transcendental equation, which we have solved numerically. In Figure 7, we have plotted $\tilde{Q}_{M+N}(M_{opt})$ as a function of $R\Delta$. By investigating the integer values of M that bracket the solution of (21), we can then find the best M+N code, i.e., the best M-ary code with N replacement signals such that $M+N = \lceil \tilde{Q}_{M+N}(M_{opt}) \rceil$. Here "best" again means "maximum recycle time, $NT = \left\{ \left\lceil \frac{R\Delta}{\log_2 M} \right\rceil - 1 \right\} \frac{\log_2 M}{R}$." For example, when $R\Delta = 4$ the solution of (21) is $M \cong 2.70$, for which $\dot{Q}_{M+N} = 4.49$. For M = 2, the recycle time is 3/R. For M = 3, the recycle time is 3.17/R, whence the (3+2) code is judged the better. We have shown the best codes for each integer $R\Delta$ in Figure 7.

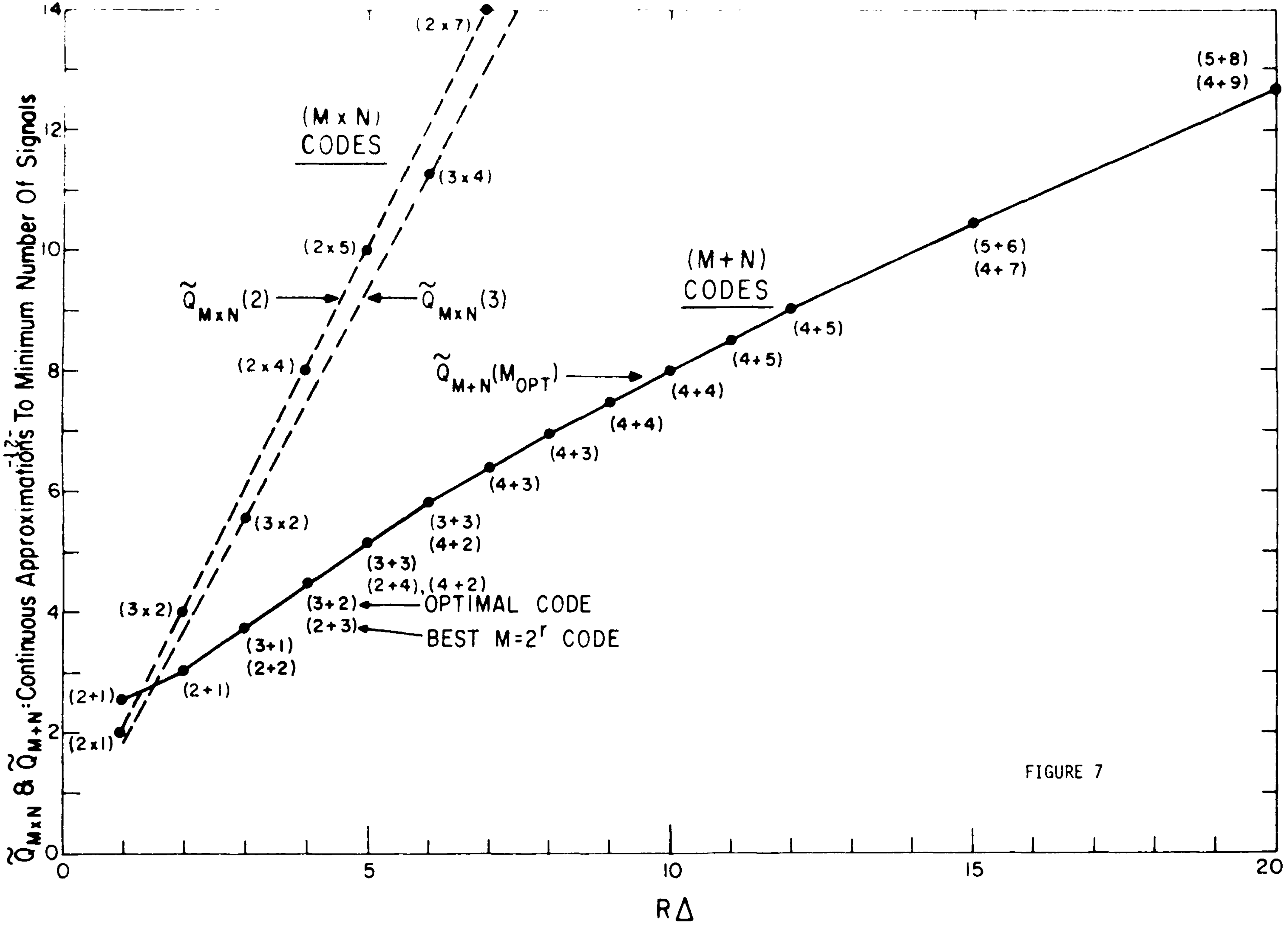

$\tilde{Q}_{M\times N}$ & $\tilde{Q}_{M+N}$: Continuous Approximations To Minimum Number Of Signals
RΔ
0
2
4
6
8
10
12
14
0
5
10
15
20
(M x N)
CODES
(M+N)
CODES
$\tilde{Q}_{M\times N}(2)$
$\tilde{Q}_{M\times N}(3)$
$\tilde{Q}_{M+N}(M_{OPT})$
(2x1)
(2+1)
(3x2)
(2+1)
(3+1)
(2+2)
(3x2)
(2x4)
(2x5)
(3x4)
(2x7)
(3+2)
(2+3)
(3+3)
(2+4), (4+2)
OPTIMAL CODE
BEST M=2^r CODE
(3+3)
(4+2)
(4+3)
(4+3)
(4+4)
(4+4)
(4+5)
(4+5)
(5+6)
(4+7)
(5+8)
(4+9)

FIGURE 7

Note that the best M+N codes do not always have $M = 2^r$, r an integer. Since $M = 2^r$ is often desirable, we have also shown in Figure 7 the best such code(s). For example, for $R\Delta = 4$, the recycle times for M = 2, N = 3 and for M = 4, N = 1 are 3/R and 2/R, respectively, so the (2+3) code is listed under the optimal (3+2) code as the best (2^r+N) code. On the other hand, for $R\Delta = 5$, the optimal (3+3) code has a recycle time of 4.75/R, while the (2^r+N) codes (2+4) and (4+2) both have recycle times 4/R; both are shown in Figure 7.

Note that MxN codes generally require very many more signals than M+N codes; in our simulation experiments we restricted ourselves to the latter.

3.3. Performance Analyses

3.3.1. MxN Codes

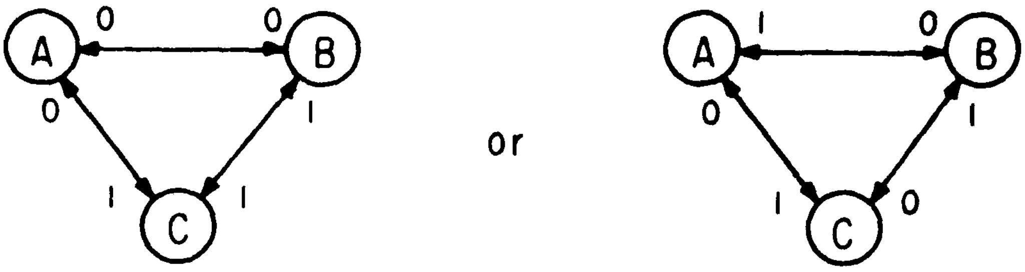
It has been shown in [8] that coherent reception of the binary-PSK 2xN code set $\{\pm S_i\}_{i=1}^{N}$, where S_i and S_j are uniformly orthogonal $(i \neq j)$, leads to approximately the same performance for RAKE as for a fixed binary PSK code set. One can guess that the same result holds for DRAKE. These results hold for either $\Delta < T$ or $\Delta >> T$, but in the latter case channel estimation is vastly simpler for 2xN codes than for fixed binary codes.

As discussed in [1], the urban mobile channel is not amenable to coherent reception, and we there resorted to differentially coherent reception with DPSK. However, commutation codes will not support DPSK, so we must now change to orthogonal keying. We can therefore conjecture that use of orthogonal keying with 2xN codes will lead to a performance about 3 dB worse than the DPSK results of [1], for both $\Delta < T$ and $\Delta >> T$.

On the other hand, PDI reception will not work at all for a fixed code set when $\Delta >> T$. Commutation codes allow use of PDI when $\Delta >> T$. We conjecture that orthogonal 2xN codes with PDI will lead to performance about 3 dB worse than the $\Delta < T$ PDI results of [1], and somewhat worse than this for $\Delta >> T$.

3.3.2. M+N Codes

The situation with M+N codes is much more complicated. As previously indicated, errors occur not only because decisions on current signals may be in error, but also because errors on previous signals may propagate, i.e., cause errors in the current signal-to-symbol decoding table.



(2+1) code no. 1

(2+1) code no. 2

FIGURE 8

Consider, for example, a (2+1) code, consisting of three signals A, B and C. Signal-to-symbol relationships can be shown in a state diagram as in Figure 8. Note that in neither of these codes can a signal follow itself. However, code #1 seems to have an advantage over code #2 since, in the former, reception of signal A or C results in an output 0 or 1 respectively, independently of whether an error occurred on the decision concerning the previous signal. That is, if (say) AC is sent and BC is decided upon at the receiver, an error may have occurred in decoding B, but not in decoding C. On the other hand, in code #2, errors can always propagate. The AC → BC error just discussed can cause errors not only in the decoding of the erroneous decision B, but also in the decoding of the correct decision C.

Analysis of M+N codes -- i.e., evaluation of their error probability performances -- is extremely complicated, and involves such questions as:

1. Are cyclically symmetric (e.g., (2+1) code #2 above) or asymmetric codes ((2+1) code #1) better?

2. Although no signal can be repeated for $N = \lceil\frac{\Delta}{T}\rceil - 1$ transmissions, is it better to allow decision among the complete signal set or not? For example, in (2+1) code #1, suppose A is sent and C detected in error. On the next transmission, an M+N receiver might be constrained to look only at the outputs of the detectors matched to B and A, since the receiver assumes that C cannot happen again. If C is now actually sent (corresponding to sequence AC or data bit 1), the receiver will have to decide between matched filter outputs A and B, neither having signal, leading to a probability of error close to

$\frac{1}{2}$. (In fact, in the case of multipath, since detector A is still responding to the previous transmission of A, the present decision will be biased toward A or a data bit 0; this will make the error probability greater than $\frac{1}{2}$.)

On the other hand, if the receiver never excludes a signal from consideration, it always makes a ternary decision (in the (2+1) case) rather than a binary decision. In the previous example, when AC is sent and C? is to be decided, the receiver decides among A, B and C on the second transmission. Since both detectors A and C will have signal (in the multipath case, where detector A is still responding to the previous signal), the choice will be largely between these two, but C is highly likely to win, leading to output data bit 1.* Thus, in this example (AC→CC), error recovery is afforded with high probability. The question is whether more or fewer errors occur on the average using ternary rather than binary decisions in the (2+1) code case (M+N'ary rather than M'ary in general). The question is greatly complicated by the nature of the multipath channel and the choice of detector (RAKE or PDI).

The questions just outlined are complex and will likely not yield to analysis when a realistic channel model is assumed. We show in Section 4 the results of a set of simulation experiments that definitively answer the questions. In the remainder of the present section, we analyze the (2+1) codes of Figure 8 for the one-path additive Gaussian channel, so as to try to obtain some insight into questions 1 and 2.

3.3.3. Simplified Analysis of the (2+1) Code

We first analyze (2+1) code #1. Here we can distinguish three cases:

Case 1: Binary decisions between the two signals that were not decided on in the previous interval (i.e., if A is the previous decision, the present decision is between B and C).

Case 2: Ternary decisions with BB called an error.*

Case 3: Ternary decisions with BB always decoded as a 1.*

*A convention must be adapted that a detected sequence AA is decoded as 0, CC as 1, and BB either as 0 or 1.

Case 1:

We first calculate the probability P_{ES} that an isolated signal is erroneously detected. There are three ways of making such an error. Thus

P_{ES} = Pr[previous decision correct] Pr[present decision erroneous]
+ Pr[previous decision erroneous and is present signal]
+ Pr[previous decision erroneous and is not present signal]
·Pr[present decision erroneous] (24)

Assume, in evaluating each term, that a present error is independent of the previous transmission;* e.g., assume a single-path channel with additive white Gaussian noise. Then (24) becomes

$$P_{ES} = (1-P_{ES})P_E^{(2)} + P_{ES}\cdot\frac{1}{2} + P_{ES}\cdot\frac{1}{2}\cdot P_E^{(2)} \tag{25}$$

where $P_E^{(2)}$ is the probability of error on a binary decision between two signals, one of which is actually present. Solving for P_{ES}:

$$P_{ES} = \frac{2P_E^{(2)}}{1+P_E^{(2)}} \xrightarrow[P_E^{(2)}\to 0]{} 2P_E^{(2)} \tag{26}$$

P_{ES} is the probability of an erroneous decision on a signal. We now must calculate the probability of erroneous decoding of a symbol, based on the decoding graph #1 of Figure 8; this is the probability of error on the present bit. There are three ways of erroneously decoding a bit.

P_{EB} = Prob(NE, prev. sig.) Prob(E, pres. sig.|NE, prev. sig.) (27)
·Prob(no decoding correction | NE, prev → E, pres.)
+ Prob(E, prev.) Prob(NE, pres. | E, prev.) Prob(no corr.|E → NE)
+ Prob(E, prev.) Prob(E, pres. | E, prev.) Prob(no corr.|E → E)

where E = error, NE = no error.

*I.e., no intersymbol interference.

We have:

$$\text{Prob(NE, prev. sig.)} = (1-P_{ES}) \tag{28}$$

$$\text{Prob(E, pres.}|\text{NE, prev.)} = P_E^{(2)} \tag{29}$$

$$\text{Prob(no decoding correction}|\text{NE, prev.; E, pres.)} = 1 \tag{30}$$

$$\text{Prob(E, prev.)} = P_{ES} \tag{31}$$

$$\text{Prob(NE, pres.}|\text{E, prev.)} \tag{32}$$

$$= \text{Prob(NE, pres.}|\text{E, prev. = pres. sig.) Pr(E, prev. = pres. sig.)}$$

$$+ \text{Prob(NE, pres.}|\text{E, prev.} \neq \text{pres. sig.) Pr(E, prev.} \neq \text{pres. sig.)}$$

$$= 0\cdot\frac{1}{2} + (1-P_E^{(2)})\cdot\frac{1}{2} \tag{33}$$

$$\text{Prob(no dec. corr.}|\text{E} \rightarrow \text{NE)} = \frac{1}{3} \tag{34}$$

since from diagram #1 of Figure 8, if the present signal is received correctly and is A or C, the present bit is decoded correctly; while, if a present B is received correctly and prev. sig. is received incorrectly, no decoding correction will take place.

We also have*

$$\text{Prob(E,pres.}|\text{E, prev.)}$$

$$= \text{Prob(E,pres.}|\text{E,prev. = pres. sig.) Pr(E, prev. = pres. sig.)}$$

$$+ \text{Prob(E, pres.}|\text{E, prev.} \neq \text{pres. sig.) Pr(E, prev.} \neq \text{pres. sig.)}$$

$$= 1\cdot\frac{1}{2} + P_E^{(2)}\cdot\frac{1}{2} \tag{35}$$

$$\text{Prob(no. dec. corr.}|\text{E} \rightarrow \text{E)} = 1 - \text{Prob(dec. corr.}|\text{E} \rightarrow \text{E)}$$

$$= \frac{7}{12} \tag{36}$$

*See Appendix.

Using (26) and (28)-(36) in (27), we finally have

$$P_{EB} = (1-P_{ES})P_E^{(2)} + P_{ES}(1-P_E^{(2)})\cdot\frac{1}{2}\cdot\frac{1}{3} + P_{ES}\left(\frac{1+P_E^{(2)}}{3}\right)\cdot\frac{7}{12}$$

$$= \frac{1-P_E^{(2)}}{1+P_E^{(2)}}\, P_E^{(2)} + \frac{2P_E^{(2)}}{1+P_E^{(2)}}\,(1-P_E^{(2)})\cdot\frac{1}{6} + \frac{2P_E^{(2)}}{1+P_E^{(2)}}\,\frac{1+P_E^{(2)}}{2}\,\frac{7}{12}$$

$$= P_E^{(2)}\left[\frac{4}{3}\left(\frac{1-P_E^{(2)}}{1+P_E^{(2)}}\right) + \frac{7}{12}\right] \xrightarrow[P_E^{(2)}\to 0]{} \frac{23}{12}\,P_E^{(2)} \qquad (37)$$

Case 2:

Here, ternary decisions are made, so AA, BB, and CC are possible decision sequences. For code #1, AA → 0, CC → 1, and we call BB an error. The probability of error P_{ES} of an isolated signal is now just $P_E^{(3)}$, the probability of error on a ternary decision among three signals, one of which is present. The probability of bit error is still given by (27), but now errors on successive signals are independent, e.g., Prob (E, pres.|NE, prev.) = Prob(E, pres.) = $P_E^{(3)}$. It is simple to show that (27) becomes

$$P_{EB} = (1-P_E^{(3)})\, P_E^{(3)}\cdot\frac{5}{6} + P_E^{(3)}\,(1-P_E^{(3)})\cdot\frac{1}{3} + [P_E^{(3)}]^2\cdot\frac{2}{3} \qquad (38)$$

In (38), the factor of $\frac{1}{3}$ in the second term is that given by (34); the factor of $\frac{2}{3}$ in the third term is given in the Appendix; and the factor of 5/6 in the first term comes from the calculation

$$\text{Prob(no dec. corr.}|\text{NE} \to \text{E}) \approx 1 - \text{Prob(dec. corr.}|\text{NE} \to \text{E})$$

$$= 1 - \text{Prob(prev. sig. = A (or C)) Prob(pres. sig. = B)}$$

$$\cdot\text{Prob(pres. decision = A (or C)}|\text{pres. decision erroneous)}$$

$$= 1 - \frac{2}{3}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{5}{6} \qquad (39)$$

where, as before, we assume no intersymbol interference, e.g., a single-path channel with additive white Gaussian noise.

Manipulating (38), we have

$$P_{EB} = \frac{7}{6} P_E^{(3)} - \frac{1}{2} [P_E^{(3)}]^2 \xrightarrow[P_E^{(3)} \to 0]{} \frac{7}{6} P_E^{(3)} \qquad (40)$$

Case 3:

Case 3 differs from Case 2 only in always decoding BB as a 1 instead of calling it an error. We have:

$$\Pr(\text{no dec. corr.}|NE \to E) = 1 - \frac{1}{6}^{*} - \Pr(\text{prev. sig.} = B)\Pr(\text{pres. sig.} = C)$$

$$\cdot \Pr(\text{pres. decis.} = B|\text{pres. decis. erron.})$$

$$= 1 - \frac{1}{6} - \frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{3}{4} \qquad (41)$$

$$\Pr(\text{no dec. corr.}|E \to NE) = 1 - \frac{2}{3}^{**} - \Pr(CB \to BB|\text{prev. decis. erron.})$$

$$= 1 - \frac{2}{3} - \frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{4} \qquad (42)$$

$$\Pr(\text{no dec. corr.}|E \to E) = \frac{15}{24}^{***} \qquad (43)$$

Then (see (38))

$$P_{EB} = (1-P_E^{(3)})P_E^{(3)}\cdot\frac{3}{4} + P_E^{(3)}(1-P_E^{(3)})\cdot\frac{1}{4} + [P_E^{(3)}]^2\cdot\frac{15}{24} \qquad (44)$$

$$= P_E^{(3)} - \frac{9}{24}[P_E^{(3)}]^2 \xrightarrow[P_E^{(3)} \to 0]{} P_E^{(3)}$$

Next we analyze (2+1) code #2. Here there are only two cases:

Case 1: Binary decisions between the two signals that were not decided on in the previous interval.

Case 2: Ternary decisions with AA, BB, and CC declared as errors.

Case 1:

Here, (30) still holds, while (34) becomes

$$\Pr(\text{no dec. corr.}|E \to NE) = 1 \qquad (45)$$

and, from the Appendix, (26) becomes

[*] As in (39).

[**] As in (34).

[***] See Appendix.

$$\Pr(\text{no dec. corr.}|E \rightarrow E) = \frac{1}{4} \tag{46}$$

(26), (27)-(29), (31)-(33) and (35) remain the same, so (37) changes to

$$P_{EB} = (1-P_{ES})P_E^{(2)} + P_{ES}(1-P_E^{(2)})\cdot\frac{1}{2} + P_{ES}\left(\frac{1+P_E^{(2)}}{2}\right)\cdot\frac{1}{4}$$

$$= P_E^{(2)}\left[2\,\frac{1-P_E^{(2)}}{1+P_E^{(2)}} + \frac{1}{4}\right] \xrightarrow[P_E^{(2)}\rightarrow 0]{} \frac{9}{4}\,P_E^{(2)} \tag{47}$$

Case 2:

Here, (39) becomes

$$\Pr(\text{no dec. corr.}|NE \rightarrow E) = 1 \tag{48}$$

while (45) holds and from the Appendix

$$\Pr(\text{no dec. corr.}|E \rightarrow E) = \frac{1}{2} \tag{49}$$

so (38) becomes

$$P_{EB} = (1-P_E^{(3)})P_E^{(3)}\cdot 1 + P_E^{(3)}(1-P_E^{(3S)})\cdot 1 + [P_E^{(3)}]^2\cdot\frac{1}{2}$$

$$= 2P_E^{(3)} - \frac{3}{2}\,[P_E^{(3)}]^2 \xrightarrow[P_E^{(3)}\rightarrow 0]{} 2P_E^{(3)} \tag{50}$$

3.3.4. Discussion

When we compare (37) with (47) and (40) with (50), we see that (2+1) code #1 is better than code #2, as expected, so we henceforth eliminate code #2 from consideration. Also, as expected, case 3 of code #1 -- eqn. (44) -- is better than case 2 -- eqn. (40). Thus the comparison is between cases 1 and 3 of code #1:

$$\begin{array}{c} (2+1) \\ \text{code \#1} \end{array} \left\{ \begin{array}{ll} \text{Binary decisions:} & P_{EB} \xrightarrow[P_E^{(2)}\rightarrow 0]{} \frac{23}{12}\,P_E^{(2)} \qquad (37) \\ & \\ \text{Ternary decisions:} & P_{EB} \xrightarrow[P_E^{(3)}\rightarrow 0]{} P_E^{(3)} \qquad (44) \end{array} \right.$$

From our discussion in Section 3.3.1, we estimate that MxN codes will operate approximately as well as similar fixed codes -- exactly as well for the simple one-path, additive-noise analysis just analyzed. Thus, for 2xN codes,

$$P_{EB} = P_E^{(2)} \tag{51}$$

A comparison of (37), (44) and (51) for orthogonal codes in the assumed one-path, additive white Gaussian noise case is shown in Table 1;* an incoherent-phase channel is also assumed. We see that the binary and ternary decision strategies behave comparably, although ternary decisions have a slight edge. (What the latter loses in having to compare three alternatives, it gains in the catastrophic case where the binary strategy would have to choose between two alternatives neither of which is correct). The 2xN codes are somewhat better, but in general require a larger total number of signals.**

TABLE I

Probability of Bit Error

$\mathcal{E}_B/N_0$	(2+1) code #1		2xN code**
	Ternary decis.	Binary decis.	(Bin. decis.)
3	≈.18	≈.19	≈.10
7	≈.03	≈.04	≈.02
10	≈.007	≈.006	≈.003

We stress that the analysis in this section is simplistic. When multipath-induced intersymbol interference is present, the (2+1) ternary strategy will probably deteriorate rapidly, at least for a PDI receiver. We discuss a less simplistic, simulation-based analysis in the next section.

*Here we have used standard formulas for this simple channel.

**A 2x2 code will suffice to suppress intersymbol interference when a (2+1) code does.

IV. Simulation Experiments

M. A. Kamil has conducted a series of simulation experiments comparing a number of M'ary systems.* In order to preserve comparability with the binary DPSK experiments reported in [1] and [2], a data rate of R = 787 kbps was set as a design parameter (the so-called "high rate" case of [1] and [2]). The systems were designed for $\Delta \approx 5\ \mu s$, typical of urban/suburban multipath. Thus $R\Delta = 4$.

4.1. Signal Structures

4.1.1. (M+N) Coding

From Figure 7, we see that the best (M+N) code is a (3+2) code.** If we want to have $M = 2^r$, we should use a (2+3) code, which has only a slightly shorter recycle time (3/R vs 3.17/R) than a (3+2) code. However, a three-replacement-symbol alphabet is relatively complex to implement, requiring a three-symbol memory at transmitter and receiver and a complicated coding/decoding table. In addition, there are major possibilities for error propagation if only a binary decision is used, while substantial additional errors are introduced if a 5'ary decision is used. We were therefore led to prefer a (4+1) code ($M = 2^2$), the (M+N) code originally proposed by Leung [7]. The recycle time of this code is only 2/R, but it would seem to be less prone to additional receiver errors whether either 4'ary or 5'ary decisions are used.

Since asymmetric codes seem better than symmetric ones (e.g., (2+1) code #1 of Figure 8 is better than (2+1) code #2), we use the coding/decoding scheme proposed by Leung in his original paper [7], as shown in Figure 9.

This code has properties similar to the (2+1) code #1 of Figure 8, e.g., correct reception of an A or an E always results in a correct decoding, regardless of whether the previous decision was correct or not. As indicated above, we test both quaternary and quinternary decision strategies. In the latter case, we follow the protocol:

AA will always be decoded as 00
BB will always be decoded as 01
CC will always be decoded as 01
DD will always be decoded as 10
EE will always be decoded as 11

*To be more fully described in Kamil's forthcoming doctoral thesis.

**The best MxN code is a 2x4 code, requiring 3 more signals than the (3+2) code and being much more complicated to realize. We therefore did not simulate MxN codes.

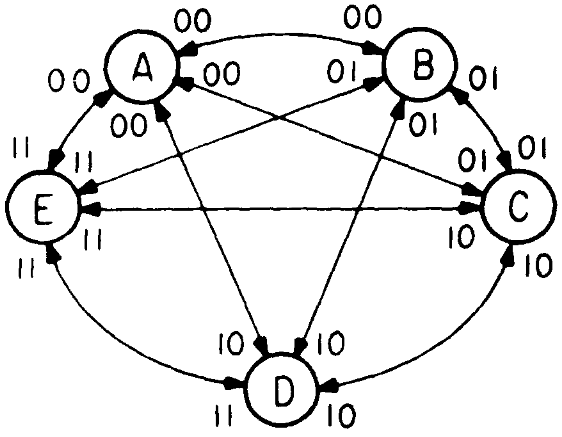


FIGURE 9

This corresponds to Case 3 of the analysis of the previous section.

The (4+1) code chosen, when run at 787 kbps, has symbol length of T = 2.54 μs. Again to preserve comparability with the previous binary experiments, we have chosen the same processing gain, TW = 127.* Thus, the basic resolution element (or "chip" in the 127-chip PN sequence we use) is 20 ns long. We therefore modified the multipath simulation package, whose resolution is 100 ns, by dividing each of its 100-ns bins into five 20-ns sub-bins and assigning paths to sub-bins in the manner described in Section VI.C of [1].

As indicated in our previous discussion, DPSK is obviously no longer applicable; we need uniformly quasi-orthogonal signals that satisfy (1) and (2). We should choose signals for which the inequality of (2) is a great as possible for all τ. Gold codes [9] seek to achieve this goal, but are more complicated than desirable for our simulations. For convenience (and, it turns out, without great loss), we therefore used

*We could equally well have preserved the previous bandwidth, W ≅ 100 MHz, leading to a processing gain TW = 254.

frequency-shifted MLSR codes of 127-chips length; these are probably not desirable for practical application because of poor data security, however.

In discrete-time form, let $\{S_k\}_1^{127}$, $S_k = \pm 1$, be a 127-chip MLSR code. Then a frequency-shifted version of this code is

$$\{S_{ik}\}_{k=1}^{127} = \{S_k e^{jki\frac{2\pi}{127}}\}_{k=1}^{127} \qquad (52)$$

where $j = \sqrt{-1}$ and $i = 1,2,\ldots,126$. We are interested in the discrete-time cross-correlation functions

$$\phi_{mn}(\ell) = \sum_{k=1}^{127} S_{mk} S^*_{n,k-\ell} \qquad (53)$$

and require, in analogy with (1), (2) and (3) that

$$\phi_{mn}(\ell) << 127 \qquad (54)$$

for all $m \neq n$ and for $\ell \neq 0$ when $m = n$.

Since $\{S_{0k}\} = \{S_k\}$ is real-valued, so is $\phi_{00}(\ell)$; and, since $\phi_{00}(\ell)$ is even, we see that its spectrum (i.e., discrete Fourier transform -DFT) is real and even around zero frequency. Further, this DFT spectrum is periodic, with period 127.

Now the DFT of $\phi_{0n}(\ell)$ is that of $\phi_{00}(\ell)$, shifted n frequency steps (n/T Hz) to the right. Since $\phi_{00}(\ell)$ is periodic and even, there are only 63 distinct shifts. Hence, there are only 63 frequency shifts of $\{S_k\}$ that yield distinct cross-correlation functions with $\{S_k\}$. Extending this reasoning and noting that (because we use envelope detectors in our receivers) only $|\phi_{mn}(\ell)|$ has significance, one can easily see that of the 127 x 126/2 pairs of signals implied in (52), there are only 64 distinct cross-correlation magnitude functions $|\phi_{mn}(\ell)|$.

Kamil has examined all of these 64 functions. For the (4+1) code, he then chose $\{S_k\}$ and four of its frequency translates, such that the six cross-correlation magnitude functions (not all necessarily distinct) are among the "best" of the 64 available functions.

4.1.2. Standard M'ary Coding

In order to mitigate intersymbol interference with standard M'ary coding, the symbol length $\log_2 M/R$ must exceed Δ. Thus, we must have

$\log_2 M \geq R\Delta = 4$ so $M \geq 16$. We chose $M = 16$ for the simulation experiments.

Again we kept the same processing gain, $TW = 127$, as in the binary experiments. In particular, Kamil has used the 127-chip MLSR code $\{S_k\}_1^{127}$ discussed above and fifteen of its frequency translates, chosen so that the corresponding 120 cross-correlation magnitude functions (not all distinct) are among the "best" of the 64 available cross-correlation magnitude functions.

For a 16'ary code run at $R = 787$ kbps, $T = \log_2 M/R = 5.08$ μs. The chip length in a 127-chip code is then 40 ns. This is not a submultiple of the 100-ns bin length of the multipath simulation package, so the sub-bin approach mentioned above (see Section VI.C of [1]) is not easily applied. To ease this problem, Kamil therefore used 50-ns chips, allowing exactly two sub-bins to a bin. This increases T to 6.35 μs (and correspondingly decreases W so that TW remains 127) and we now have $R = 1/T \cong 630$ kbps. Although the lower rate decreases intersymbol interference slightly from that at the nominal 787 kbps, it is already so small (since for both values of R, $T > \Delta$) that the system performance will surely change only negligibly.* Thus, we feel that the 630-kbps simulation results will closely represent the 787-kbps case and will be comparable to other 787-kbps results.

4.2. Receiver Structures

The basic receiver structure used in Kamil's simulation is shown in Figure 10. The received signal is passed through M' matched filters (M' = M or M' = M+N) and M' envelope detectors. Each envelope detector output is passed into a post-detection algorithm, which -- in Kamil's experiments -- was one of the following:

- PDI
- Largest-path (LP) detection**
- DRAKE
- RAKE

*We must of course maintain the same transmitted enerry per bit for the longer signals.

**See [1]. Here we examine the output of each envelope detector, choose the largest peak in each, and output the value of this peak. In our simulations, for simplicity, we actually examined the outputs of <u>all</u> detectors at the instant (known to the computer) at which the "correct" matched filter output had its largest multipath response. This was an undesirable simplification -- see below.

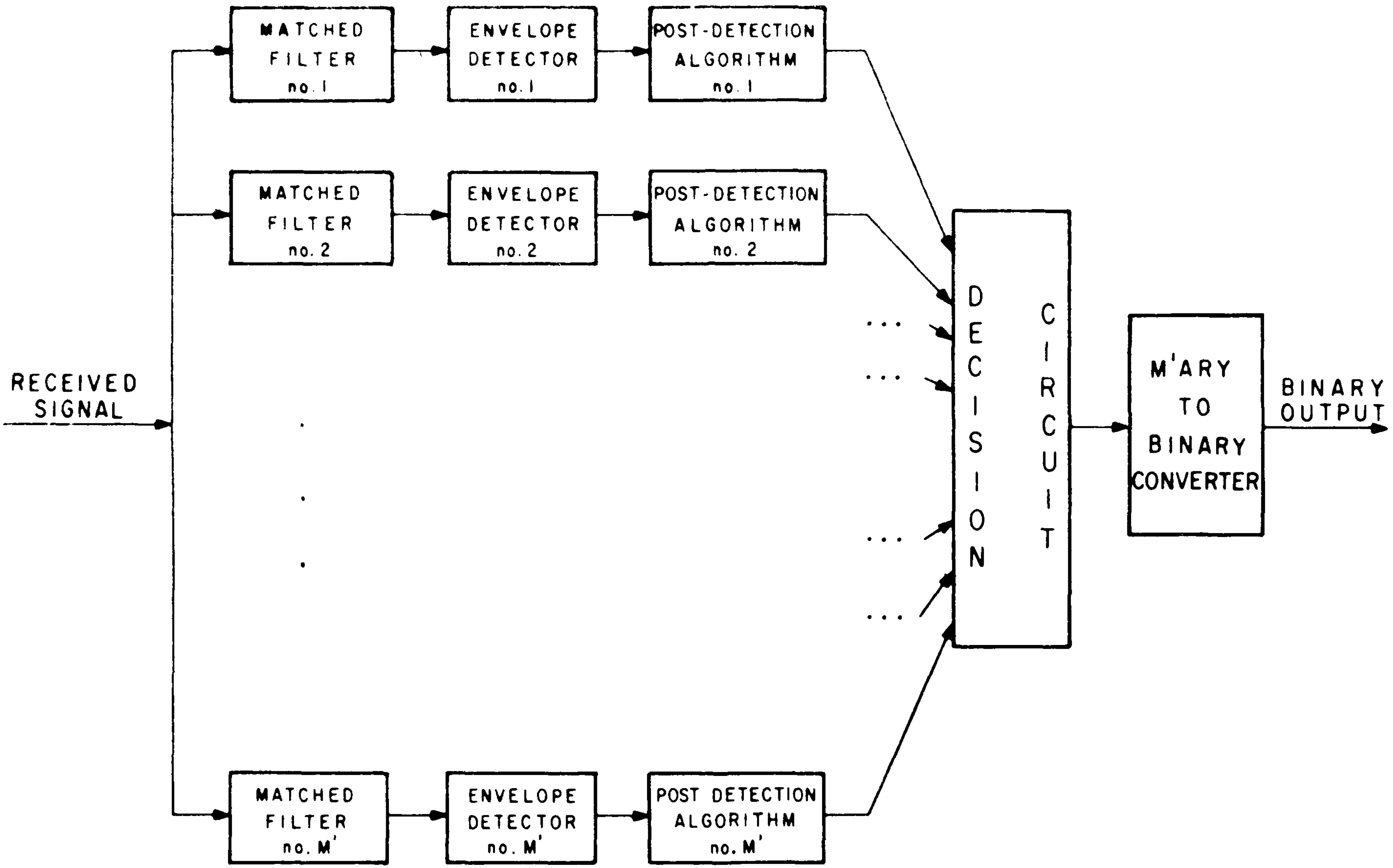

RECEIVED SIGNAL
MATCHED FILTER no. 1
ENVELOPE DETECTOR no. 1
POST-DETECTION ALGORITHM no. 1
MATCHED FILTER no. 2
ENVELOPE DETECTOR no. 2
POST-DETECTION ALGORITHM no. 2
MATCHED FILTER no. M'
ENVELOPE DETECTOR no. M'
POST DETECTION ALGORITHM no. M'
DECISION CIRCUIT
M'ARY TO BINARY CONVERTER
BINARY OUTPUT

FIGURE 10

The DRAKE and RAKE algorithms require path delay (and, for DRAKE, path strength) estimates; the PDI and LP algorithms require only bit-sync information.

The outputs of the algorithm circuits go into the decision circuit. For standard M'ary reception, this circuit identifies the largest of its inputs and passes its index to its output. If the system uses an (M+N)'ary commutation code with M'ary decision, the decision circuit looks only at the M of its inputs that are currently "possible,"[*] identifies the largest of these, and passes the corresponding index to its output. If the system uses an (M+N)'ary commutation code with (M+N)'ary decision, the decision circuit looks at all of its inputs, identifies the largest, and passes the corresponding index to its output.

For a standard M'ary system, the M'ary-to-binary convertor does a standard translation of its one-out-of-M input to $\log_2 M$-bit binary words. For an (M+N)'ary system with M'ary decision, the converter uses a decoding table for M'ary sequences such as that implicit in Figure 9. For an (M+N)'ary system with (M+N)'ary decision, the decoding table is augmented by entries for "impossible" input sequences, such as the conventions given in Section 4.1.1 for a (4+1) code.

In actuality, Kamil did not convert to binary. Instead, he compared M'ary receiver outputs (including the use of the decoding table for (M+N)'ary systems, where the decoding is M'ary to M'ary) with M'ary transmitter inputs. The linkage between M'ary probability of error and binary probability of error was made on the basis of (15).[**]

4.3. Results

The results of the simulation are shown in Figures 11-24. All of these give the bit error probability P_{EB} as a function of the average energy per bit $\bar{\mathcal{E}}_{LOS}$ received via the LOS (line of sight) path as normalized to the channel noise power density N_0.

Figures 11-17 give results for Area A, i.e., dense urban high rise (see [1]); for this area, the integration interval of the PDI receiver was taken as 4 μs. Figures 18-24 give results for Area D, suburban/residential; here, the PDI integration interval was only 1 μs since the

[*] The circuit decides which inputs are possible on the basis of its previous N decisions.

[**] It would have been better to convert all the way to binary to verify the validity of (15).

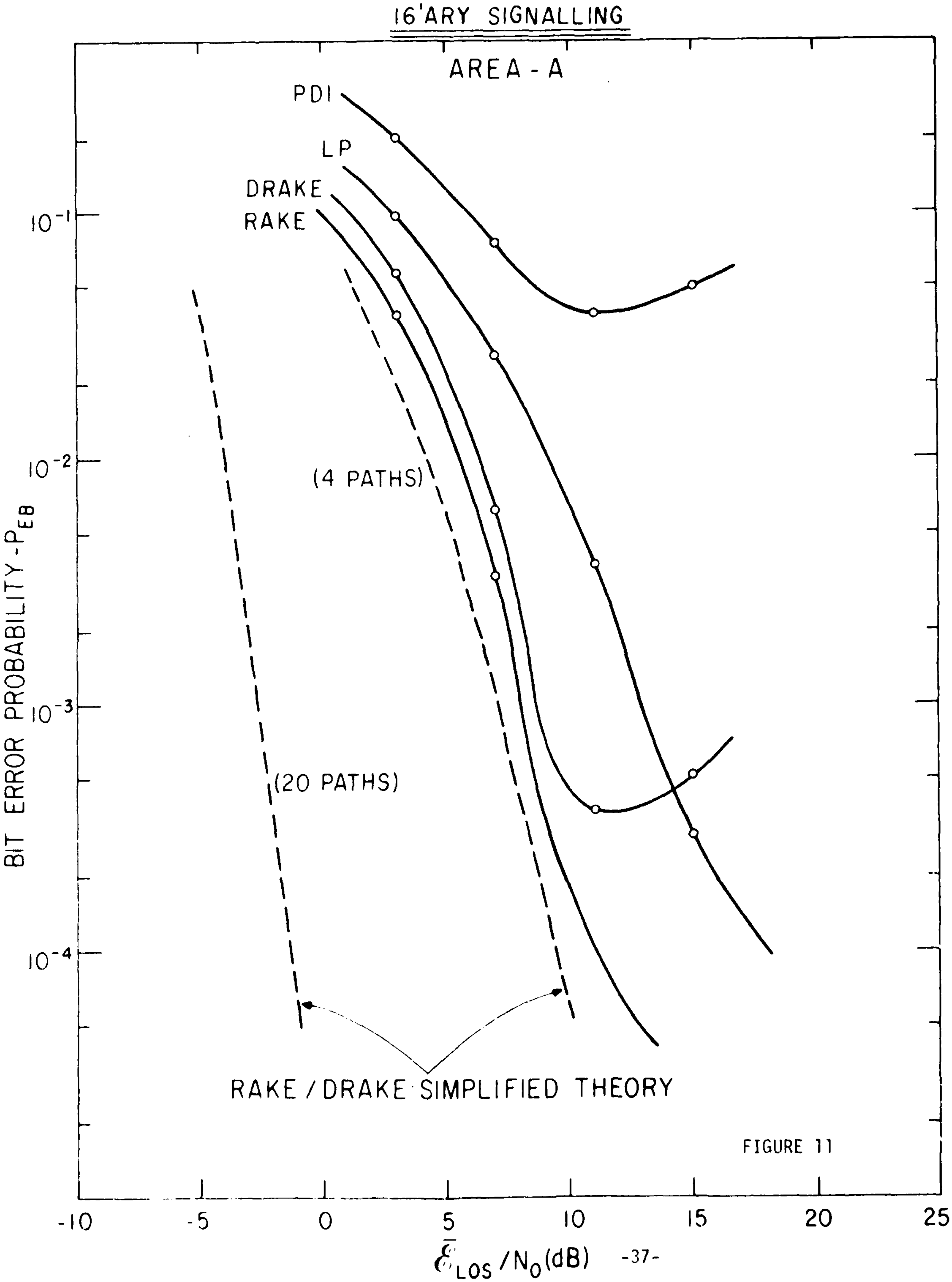

16'ARY SIGNALLING
AREA - A
PDI
LP
DRAKE
RAKE
(4 PATHS)
(20 PATHS)
RAKE / DRAKE SIMPLIFIED THEORY
BIT ERROR PROBABILITY - P_{EB}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
-10
-5
0
5
10
15
20
25
$\bar{\mathcal{E}}_{LOS}/N_0$(dB)

FIGURE 11

(4+1)'ARY SIGNALLING - 4'ARY DECISION

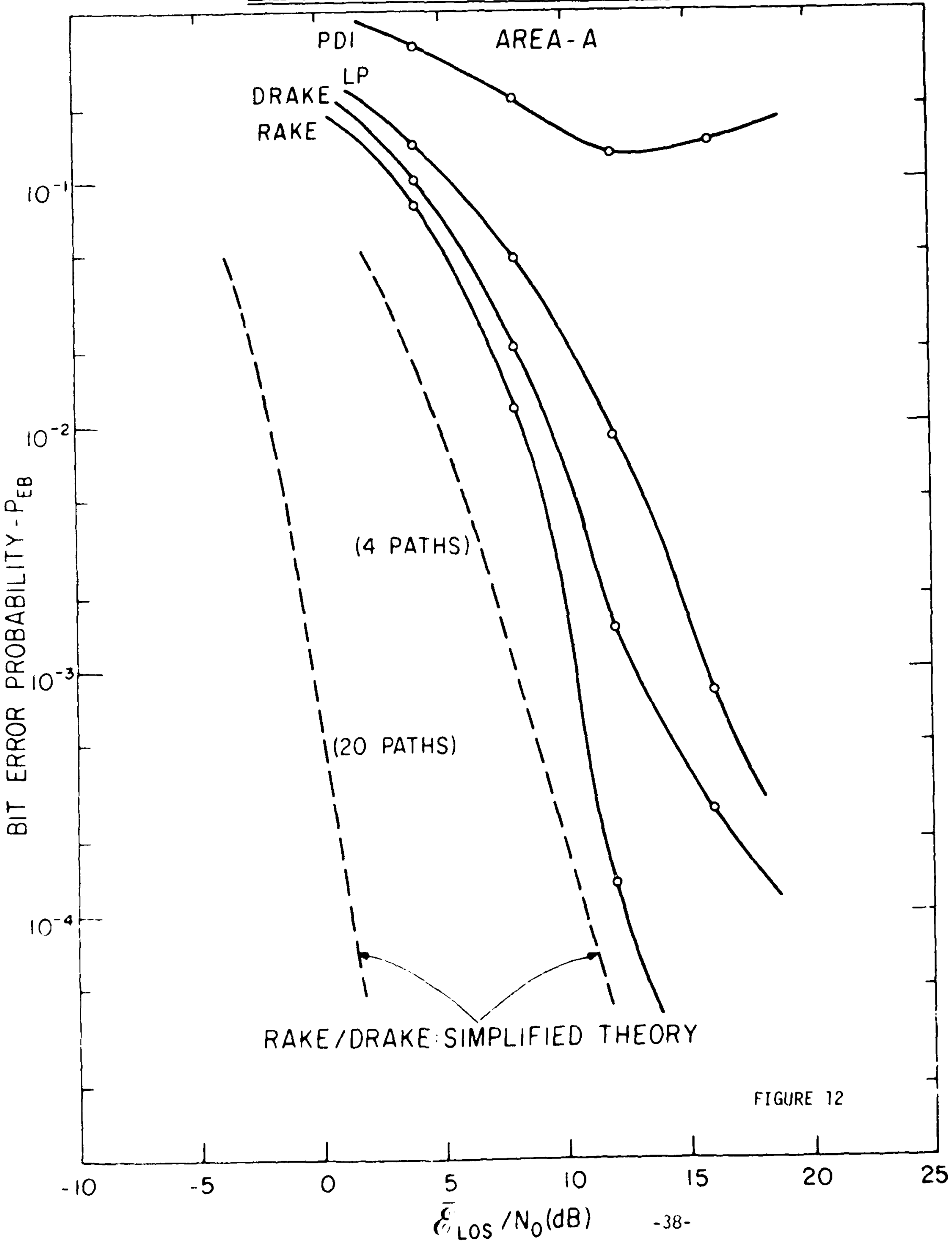


FIGURE 12

(4+1)'ARY SIGNALLING-5'ARY DECISION

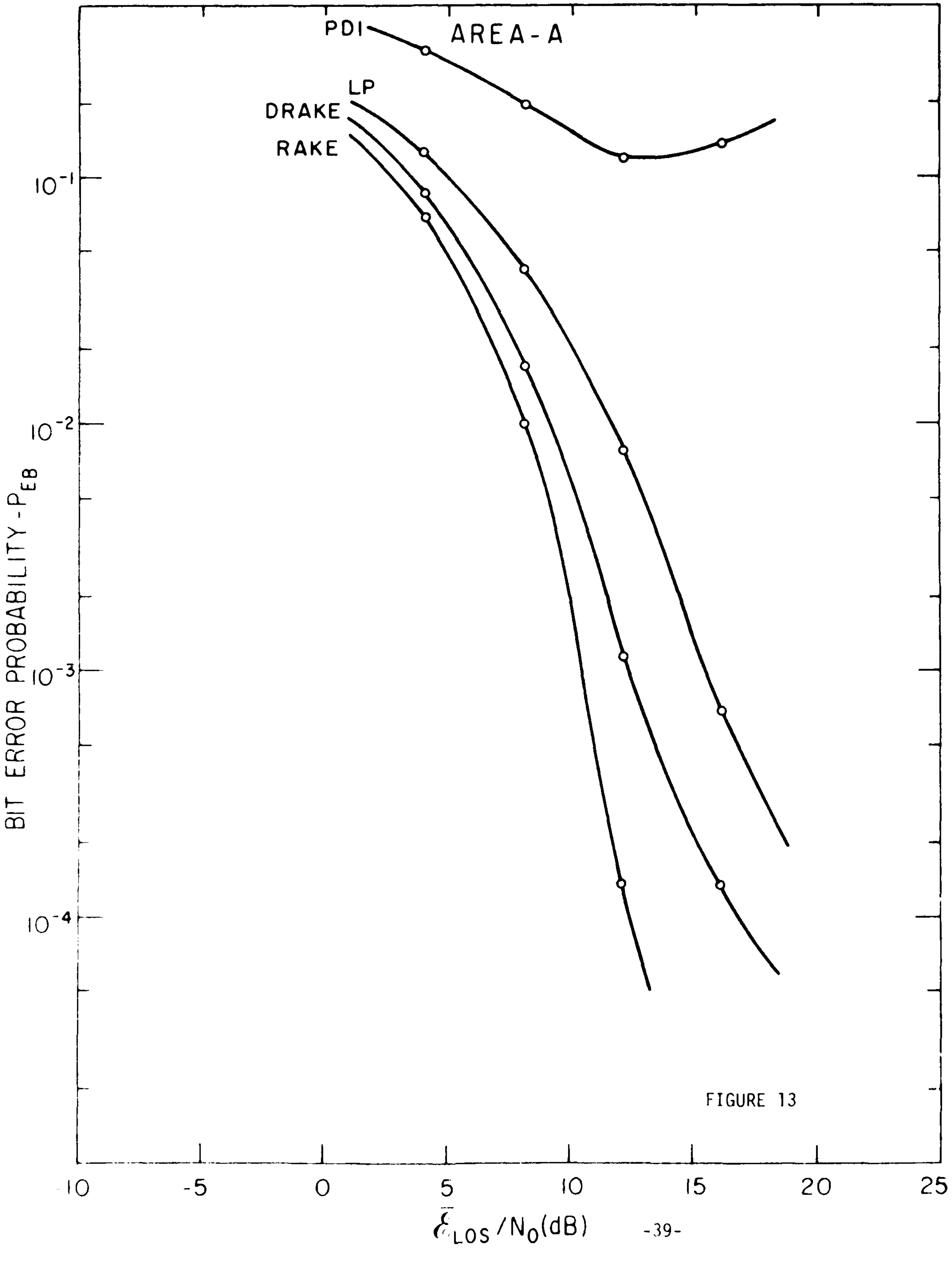


FIGURE 13

PDI RECEIVER-AREA A COMPARISON OF SIGNALLING SCHEMES

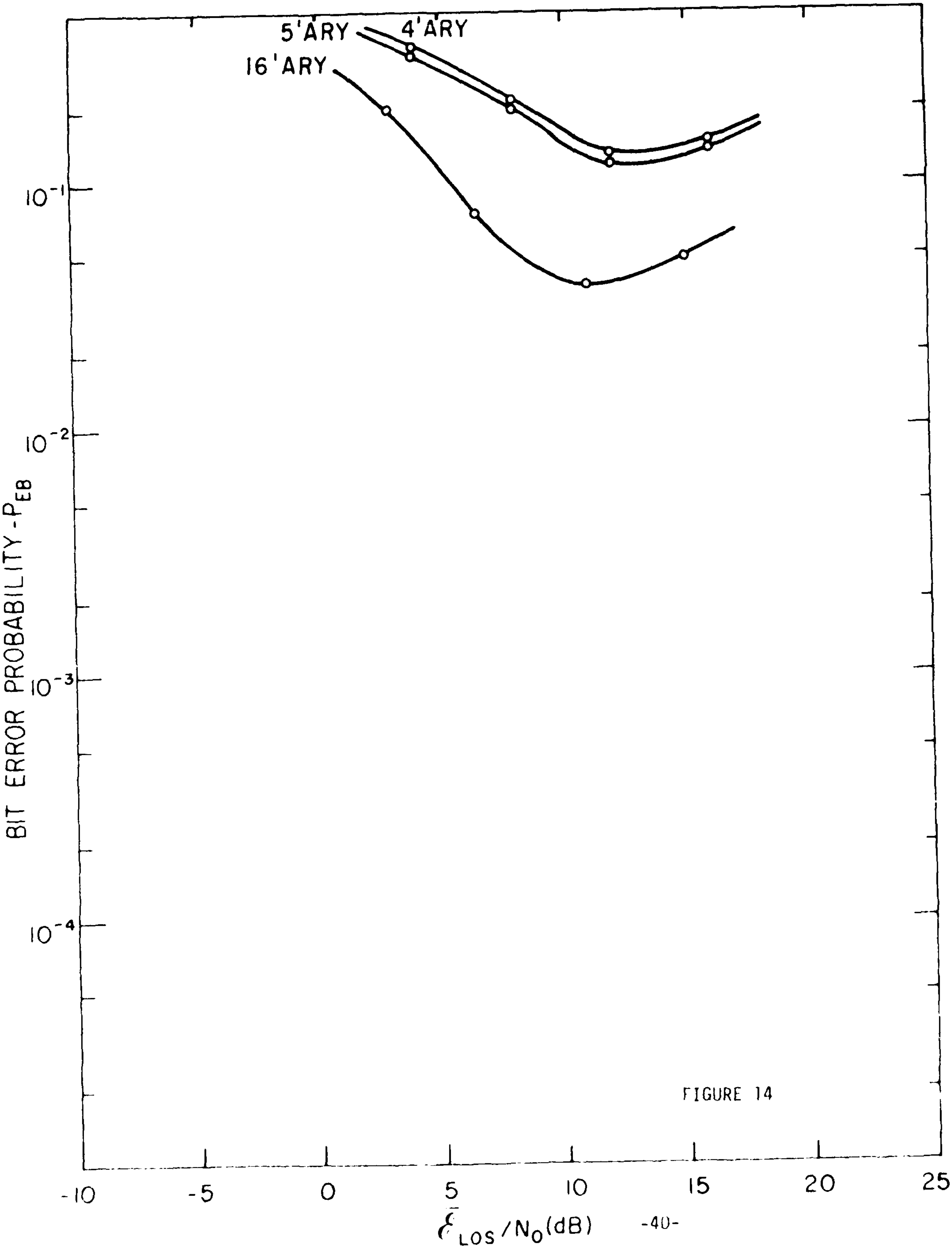


FIGURE 14

LP-AREA A

COMPARISON OF SIGNALLING SCHEMES

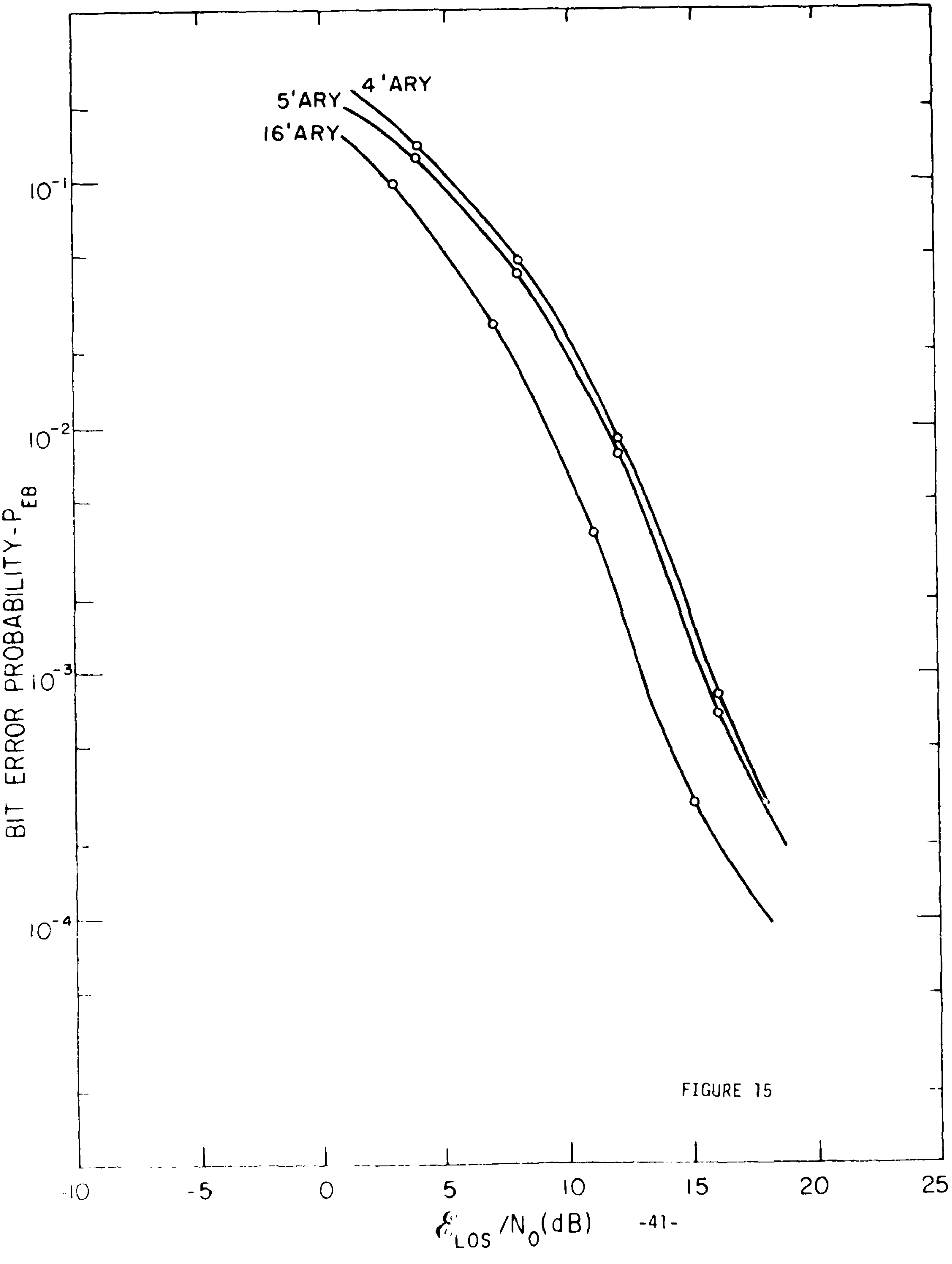


FIGURE 15

DRAKE - AREA A

COMPARISON OF SIGNALLING SCHEMES

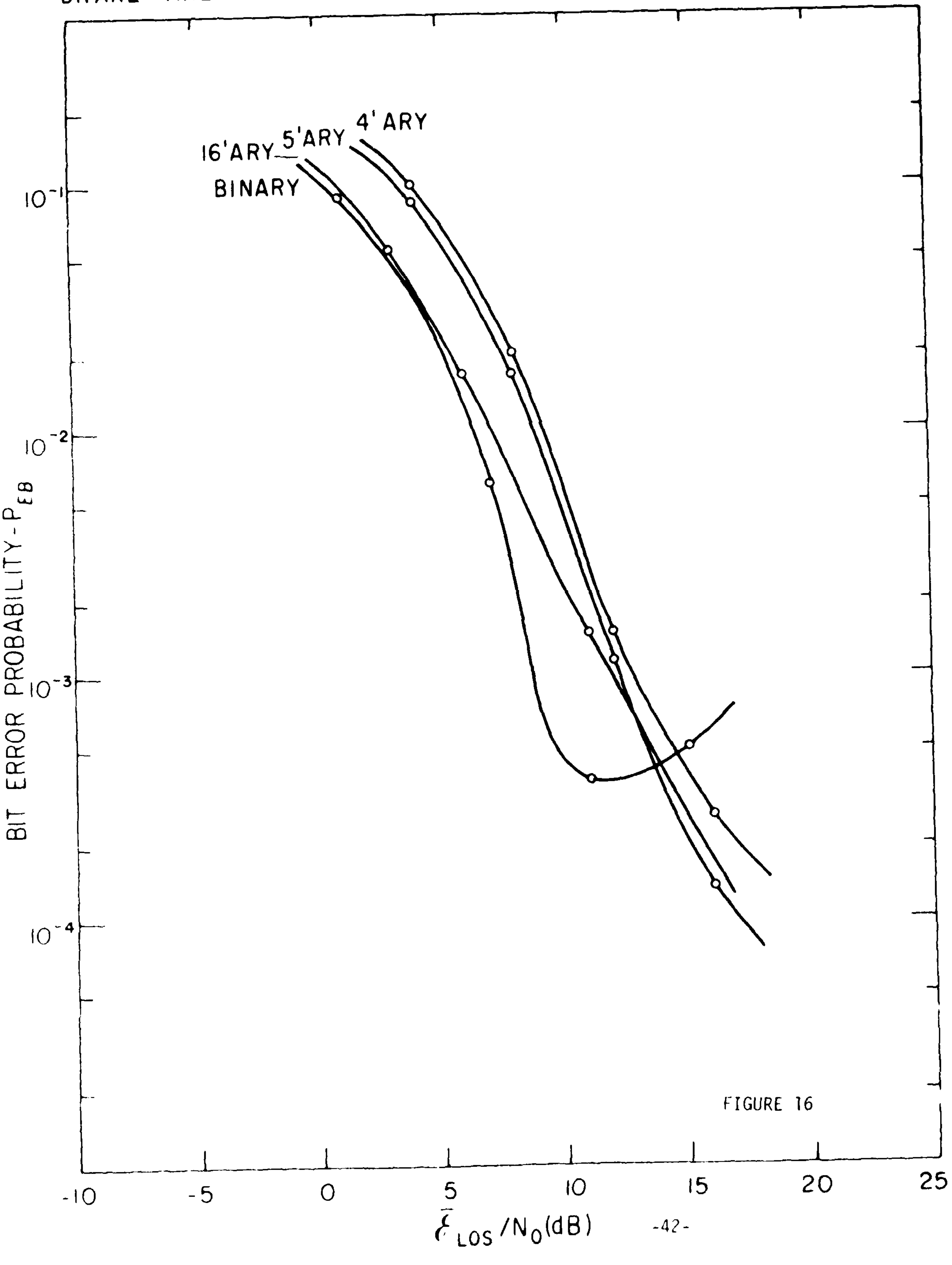


FIGURE 16

RAKE-AREA A

COMPARISON OF SIGNALLING SCHEMES

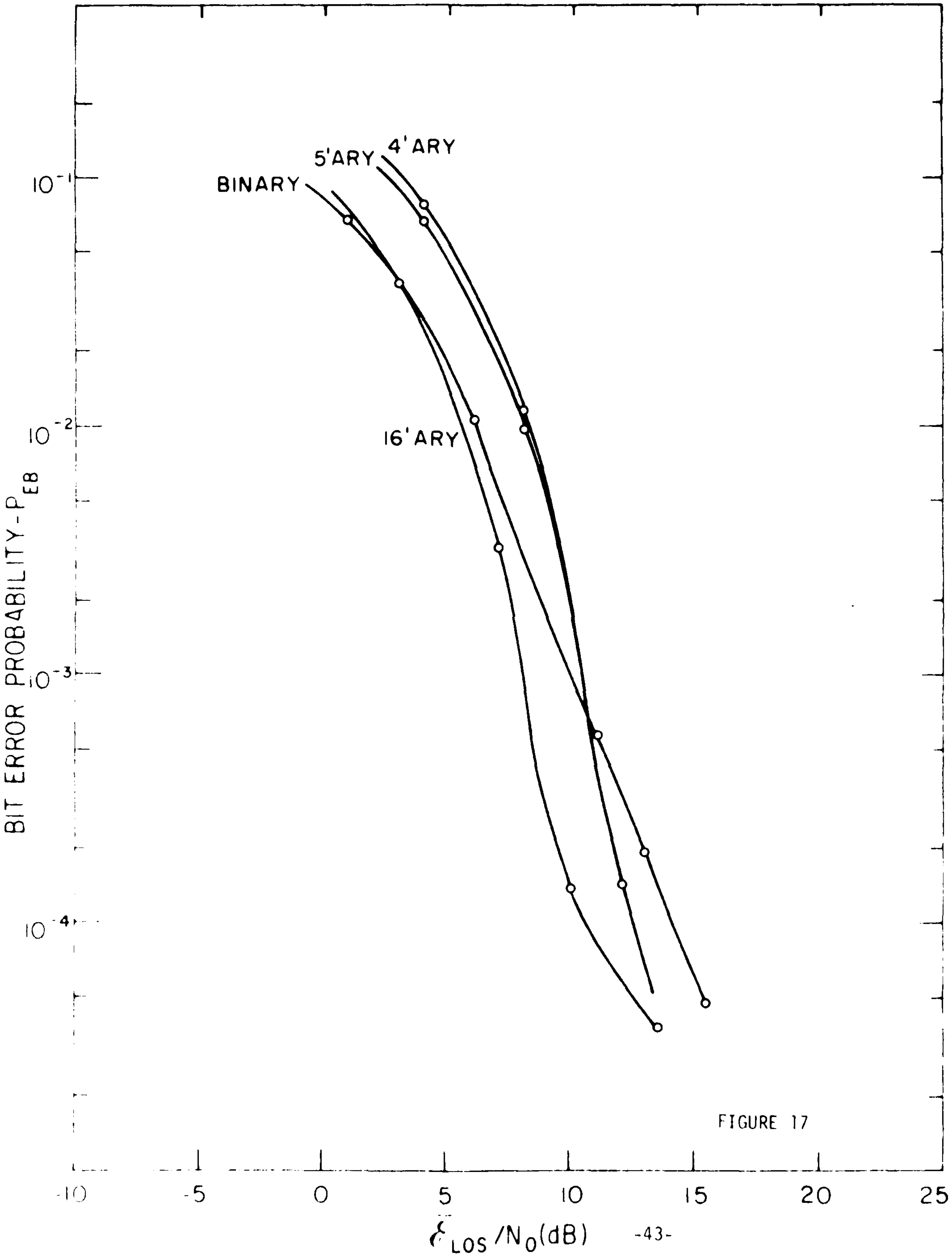


FIGURE 17

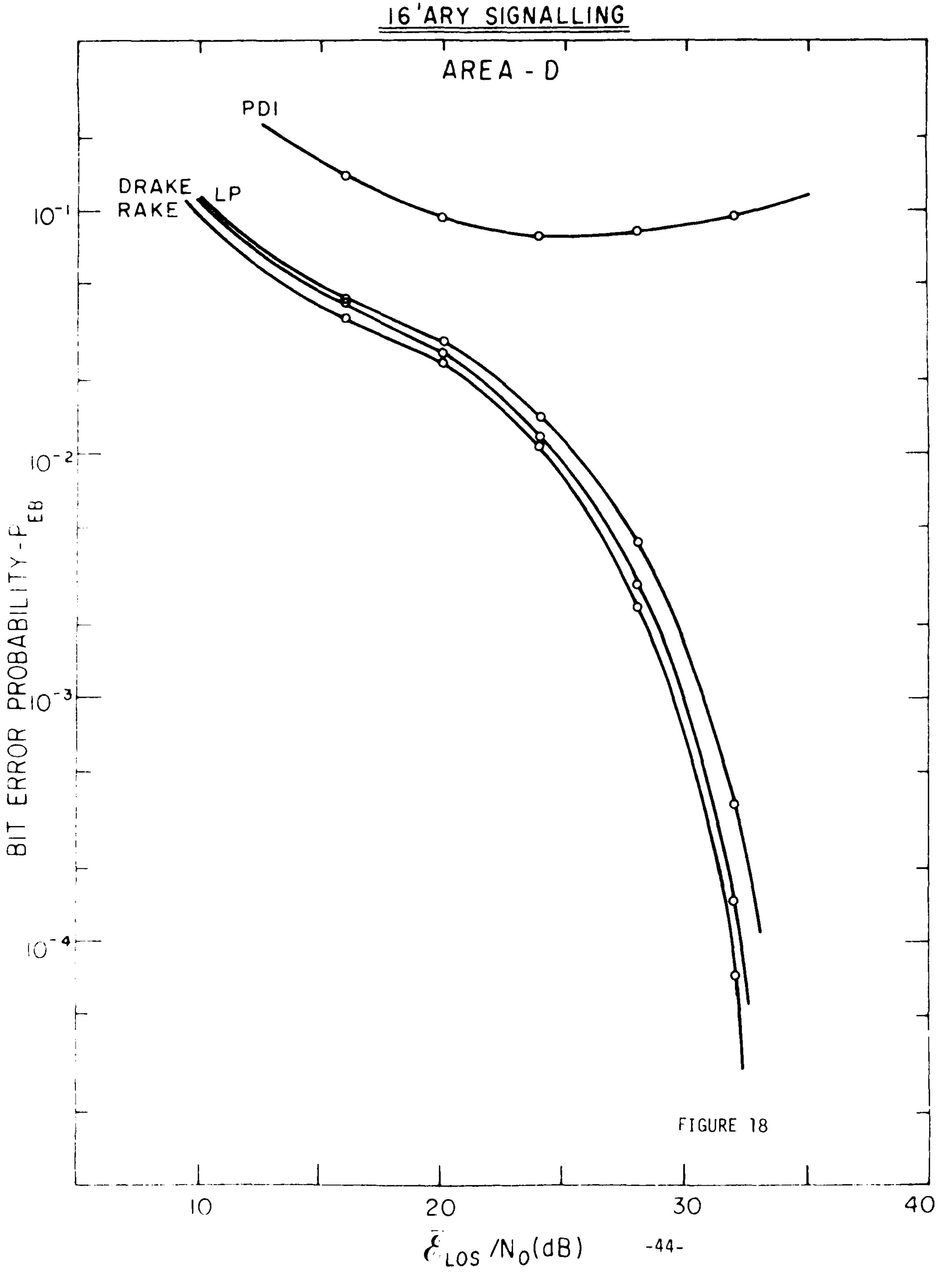

16'ARY SIGNALLING
AREA - D
PDI
DRAKE
RAKE
LP
10^{-1}
10^{-2}
10^{-3}
10^{-4}
BIT ERROR PROBABILITY - P_{EB}
10
20
30
40
$\bar{\mathcal{E}}_{LOS}/N_0$(dB)

FIGURE 18

(4+1)'ARY SIGNALLING - 4'ARY DECISION

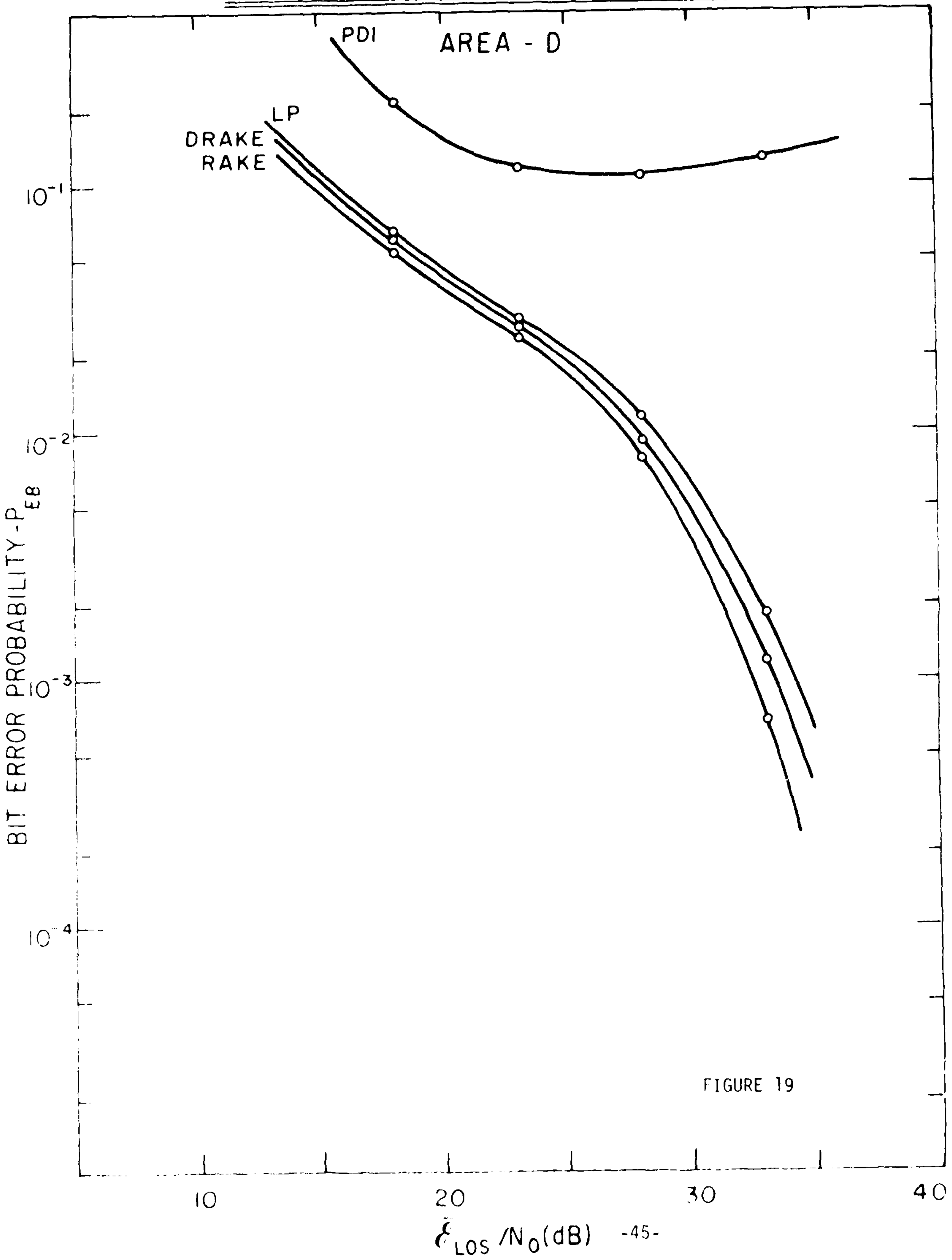


FIGURE 19

(4+1)'ARY SIGNALLING - 5'ARY DECISION

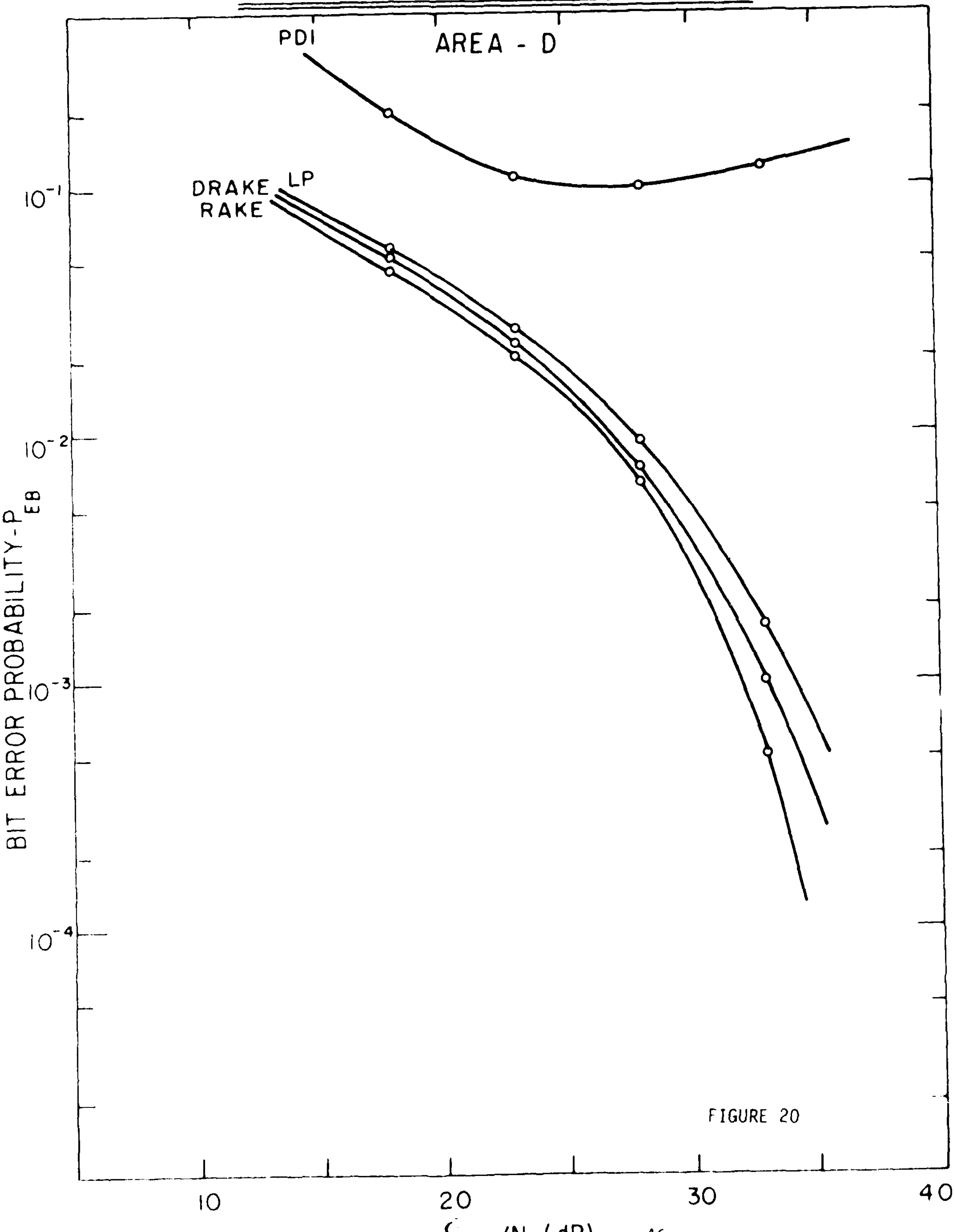


FIGURE 20

PDI RECEIVERS-AREA D COMPARISON OF SIGNALLING SCHEMES

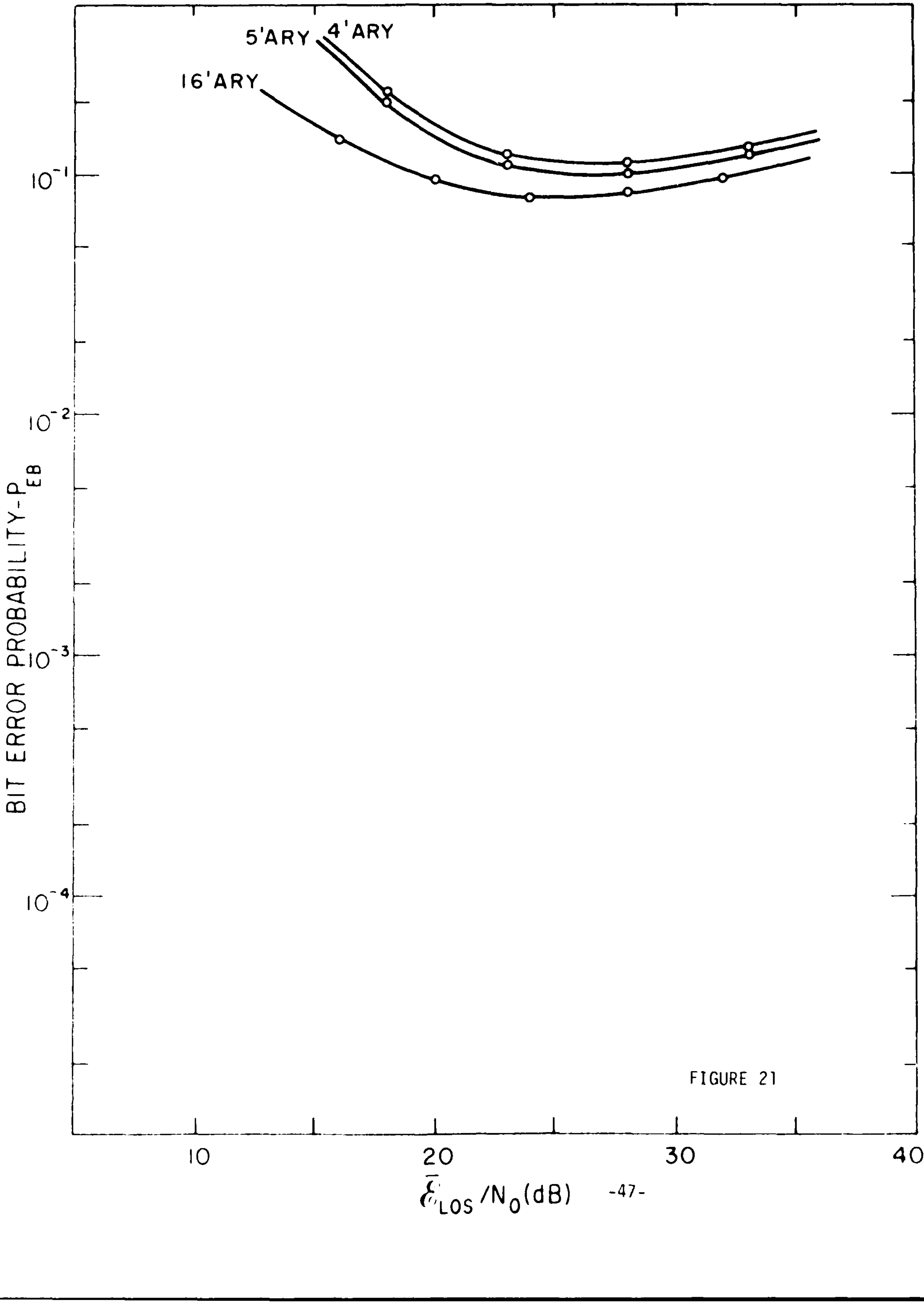


FIGURE 21

LP-AREA D

COMPARISON OF SIGNALLING SCHEMES

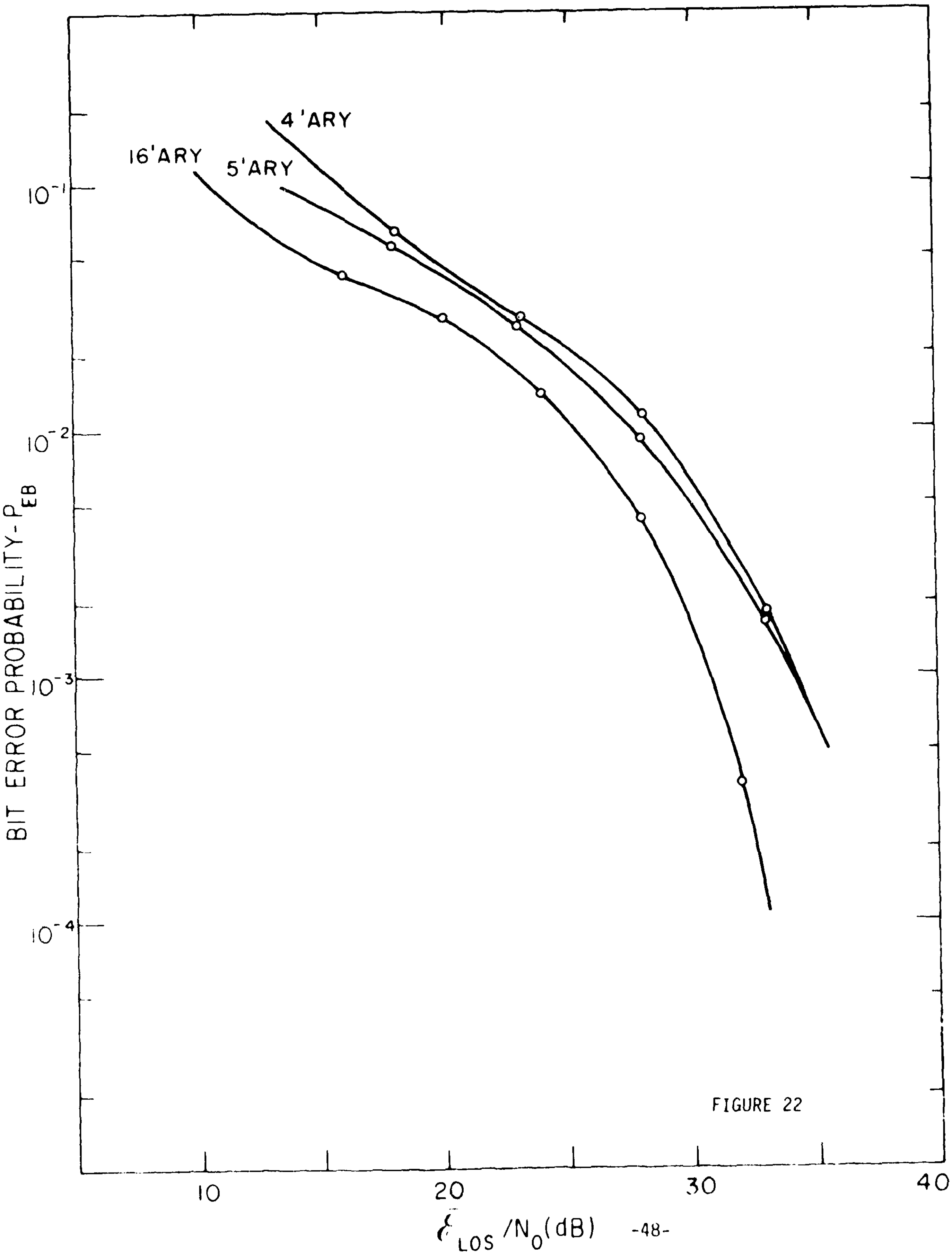


FIGURE 22

DRAKE-AREA D

COMPARISON OF SIGNALLING SCHEMES

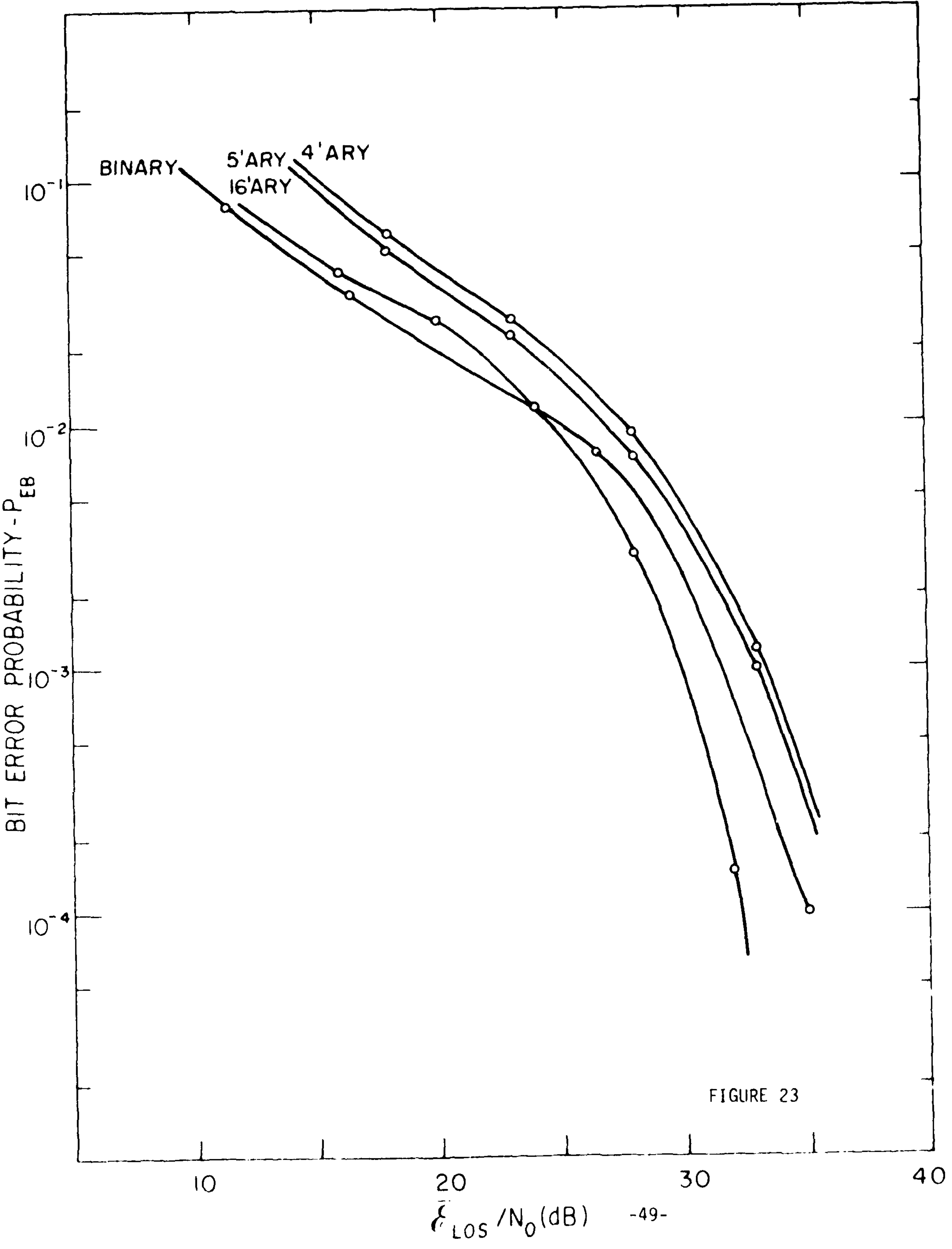


FIGURE 23

RAKE - AREA D

COMPARISON OF SIGNALLING SCHEMES

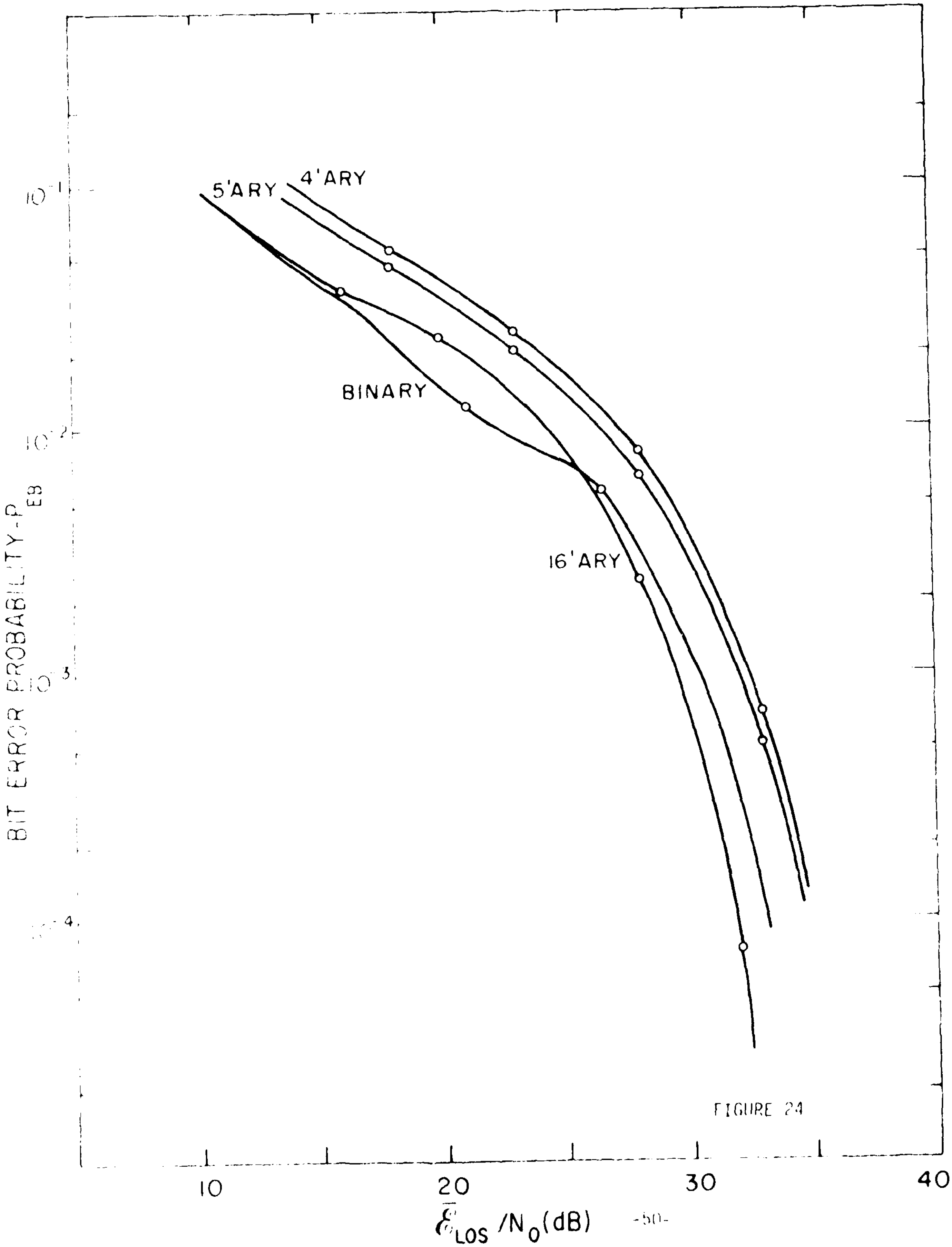


FIGURE 24

typical multipath spread is quite small.

Figure 11 shows the Area-A performance of standard 16'ary receivers. As expected, RAKE outperforms all other receivers. Importantly, and not predicted by our simplified analysis (Section 2.2.3), PDI performs very badly. Further, both it and DRAKE show a "bottoming out" behavior: above a certain $\bar{E}_{LOS}/N_0$, their performances first level off and then deteriorate. This behavior is due to the presence of crosscorrelation noise, and will be fully discussed below.

In Figure 11, we also have shown theoretical curves from Figure 5, as modified using (15) to reflect bit error probability. Recall that Figure 5 is based on a much oversimplified model involving a fixed number of independent, Rayleigh-fading paths having equal mean-square strengths and signals with zero crosscorrelations and autocorrelation sidelobes. (Under these assumptions, equal-weight combining is optimal, so DRAKE and RAKE are equivalent; the theoretical curves represent either.) By contrast, our realistic simulation program generates a random number of dependent, log-normally fading paths having unequal mean-square strengths and uses signals with substantial crosscorrelation and autocorrelation sidelobes. As we have seen before in Section VI.D of [1], the theory is far too optimistic: in Figure 11 the K = 20 theoretical curve is about 10 dB optimistic, even though the simulated multipath has an average of about 23 paths per profile. The K = 4 theoretical curve is a better fit -- i.e., ideal signals (without autocorrelation sidelobes and with zero cross-correlations) transmitted through four equal-strength, independent, Rayleigh fading paths would perform about as well as the real-life situation we have simulated.

Recall from Section 2.2.3 that we have estimated PDI to perform 2-5 dB worse than DRAKE. We see that this estimate is more or less correct for small $\bar{\mathcal{E}}_{LOS}/N_0$. But as $\bar{\mathcal{E}}_{LOS}/N_0$ becomes larger, so that crosscorrelation noise dominates channel noise (see discussion below), the estimate becomes worthless. This is because the estimate is based only on the effects of additional channel-induced noise at the receiver output caused by "turning spurious taps on;" we did not at all consider cross-correlation noise, and this type of noise becomes dominant at large $\bar{\mathcal{E}}_{LOS}/N_0$.

Figures 12 and 13 show the simulated results for (4+1)'ary signalling with 4'ary and 5'ary decisions, respectively. In Figure 12, the theoretical K = 4 curve (from Figure 3) is again much closer to real life than the K = 20 curve; K = 3 would be closer still. We have not drawn theoretical curves in Figure 13, since the theory of Section 2.2.2 does not model decision among five matched filter outputs of which only four may have signals.

Figures 14-17 compare the performances of the three different M'ary signalling schemes for each of PDI, LP, DRAKE and RAKE reception. We see that 16'ary signalling outperforms (4+1)'ary signalling in all cases, as is expected, since M'ary signalling gets more efficient as M increases. We also see that 5'ary decisions are more effective than 4'ary decisions: adding another (perhaps totally spurious) candidate to the decision algorithm more than compensates for the error propagation that would occur if that candidate were excluded. We have see this effect theoretically in Table I of Section 3.3.3.

We have also drawn in Figures 16 and 17 the curves for high-rate (787-kbps) DRAKE and RAKE reception of binary DPSK signalling, as given in [1] and [2]. (Binary PDI and LP receivers are not practicable at 787 kbps because of intersymbol interference.) Note that at small $\mathcal{E}_{LOS}/N_0$, binary DPSK is slightly better than 16'ary quasi-orthogonal keying. This is because, although 16'ary orthogonal keying is better than binary orthogonal keying, binary DPSK has approximately a 3-dB advantage over binary orthogonal keying, which compensates for the advantage of 16'ary keying. (In addition, our 16'ary keying is only quasi-orthogonal, so its performance is further deteriorated by cross-correlation noise.) As $\mathcal{E}_{LOS}/N_0$ increases, intersymbol interference -- present in the high-rate binary system, but virtually absent in our 16'ary system -- causes the binary system to lose its advantage. However, for even larger $\mathcal{E}_{LOS}/N_0$, when cross-correlation noise dominates the channel noise in 16-ary DRAKE, the binary system again becomes better. There is evidence in Figure 17 that this crossover will occur also for RAKE, as should be expected, since cross-correlation noise effects RAKE also.

Figures 18-24 repeat the curves of Figures 11-17 in the same sequence, but for Area D (suburban/residential). We have not attempted to draw theoretical curves in Figures 18 and 19 since in Area D one path usually dominates; i.e., we cannot even approximately say that there are effectively K equal-strength paths (K>1) and use Figures 3 and 5.

We see from Figures 18-20 that, as in Area A, for each signalling scheme the receivers are ranked PDI, LP, DRAKE, RAKE in order of increasingly better performance. Although the PDI curves all bottom out, the DRAKE curves do not, at least in the range of $\bar{\mathcal{E}}_{LOS}/N_0$ shown. This is because cross-correlation noise, arriving via fewer paths, is not as much a problem as in Area A; we discuss this point further below.

Figures 21-24 show that 16'ary signalling is still better than (4+1)'ary. Again, when (4+1)'ary signalling is used, 5'ary decisions are better than 4'ary decisions. In Figures 23 and 24, where high-rate binary DPSK curves from [1] and [2] are displayed, we again see that binary DPSK is better than 16'ary quasi-orthogonal keying for small $\bar{\mathcal{E}}_{LOS}/N_0$, but loses its advantage as $\bar{\mathcal{E}}_{LOS}/N_0$ increases, when intersymbol interference begins to affect it. We suspect that there is an other crossover for even larger $\bar{\mathcal{E}}_{LOS}/N_0$, as cross-correlation noise begins to dominate 16'ary reception, but the crossover is not apparent in Figures 23 and 24, since cross-correlation noise is not as important in Area D.

4.4. Discussion of Cross-correlation Noise

We now discuss the concept of cross-correlation noise more fully, since it is a major theme in understanding the performance of M'ary receivers (M>2). We have already seen in connection with Figure 1 that, when the bank of matched filters of Figure 10 is hit by the output of the multipath channel, all respond. The output of the filter matched to the transmitted signal (the "correct" filter) has a multipath peak plus autocorrelation sidelobes present for each path; each peak shows distinctly, but the sidelobes due to different paths overlap and add. The other filters respond with the cross-correlation function between the transmitted signal and the signals to which they are matched. The cross-correlation response occurs once for each path, since the transmitted signal arrives multiple times; as with the autocorrelation sidelobes in the "correct" filter's response, these cross-correlation responses overlap and add.

Figure 25 shows a computer printout of actual matched filter output envelopes from the simulation experiments. Here, the signalling is stopped toward the end, so it is possible to view only the channel noise as well as the autocorrelation noise. In Figure 25, $\bar{\mathcal{E}}_{LOS}/N_0 \cong 20$ dB.

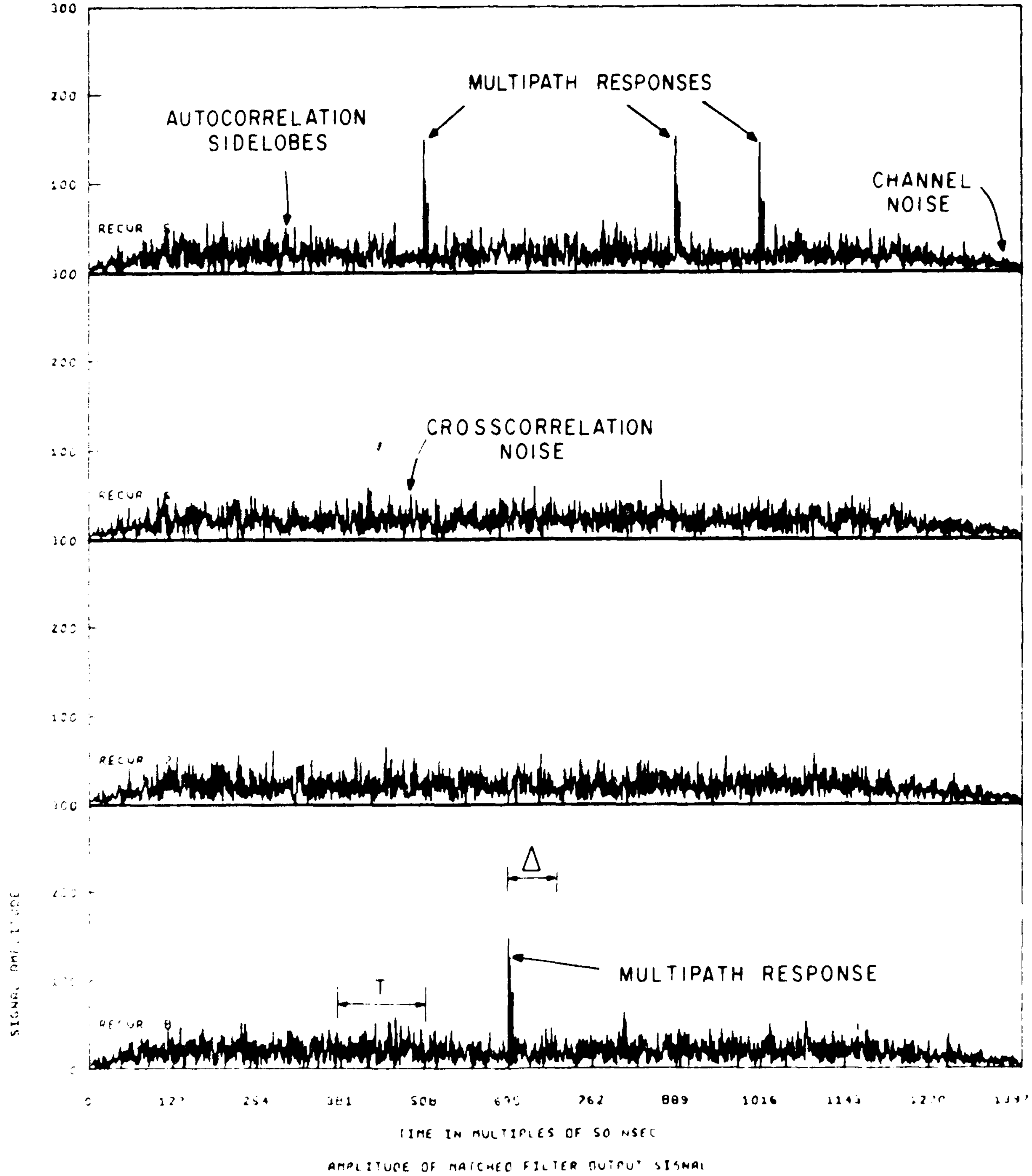

300
200
100
300
200
100
300
200
100
300
200
100
0
RECUR 5
RECUR
RECUR
RECUR 8
AUTOCORRELATION SIDELOBES
MULTIPATH RESPONSES
CHANNEL NOISE
CROSSCORRELATION NOISE
Δ
T
MULTIPATH RESPONSE
SIGNAL AMPLITUDE
0
127
254
381
508
635
762
889
1016
1143
1270
1397
TIME IN MULTIPLES OF 50 NSEC
AMPLITUDE OF MATCHED FILTER OUTPUT SIGNAL

FIGURE 25

In order to appreciate the scales of figures 1 and 25, note that if a multipath peak at the output of filter k has height 1, the rms value of its sidelobes at filter k's output will be of the order of $1/\sqrt{TW}$ ($\cong$ 0.089 or -21 dB in our case of TW = 127). The cross-correlation functions this path induces in the other filters' outputs will also have rms values of the order of $1/\sqrt{TW}$. If there are K paths of more or less equal strength, each will contribute autocorrelation sidelobes and cross-correlation functions of rms value $1/\sqrt{TW}$, and these will add rms-wise to cumulate rms totals of $\sqrt{K/TW}$.

Suppose that there are indeed K paths of more or less the same average strength as the LOS path. The rms values of the autocorrelation and cross-correlation noises in the matched filter outputs will then be down from the multipath peaks in the "correct" filter's output by a factor of the order of $\sqrt{TW/K}$. From matched-filter theory, we know that the rms value of the noise in the filter outputs, due to channel noise, will be down from the peaks by a factor of $\sqrt{2\bar{\mathcal{E}}_{LOS}/N_0}$. Then, if $2\bar{\mathcal{E}}_{LOS}/N_0 > TW/K$, the correlation noise will dominate the channel noise.

For example, consider Area A, where there is an average of about 20 paths of significant strength per profile, or the equivalent of perhaps K = 10 paths of more or less equal average strength.* For our case of TW = 127, correlation noise will dominate when $\bar{\mathcal{E}}_{LOS}/N_0 > 127/20 \cong 8$ dB. Again, in Area D, where there is the equivalent of only K = 1 equal-strength path, correlation noise will dominate when $\bar{\mathcal{E}}_{LOS}/N_0 > 127/2 \cong 18$ dB.

Indeed, we see from Figures 14 and 21 that PDI performance begins to "bottom out" -- i.e., the curves start to have an upward curvature -- at $\bar{\mathcal{E}}_{LOS}/N_0 = 8$ and 18 dB respectively in Areas A and D. That is, above these values of SNR, further decreases in channel noise will become less and less significant as the constant-level correlation noise more and more controls performance. In fact, we see that there is a slight deterioration of performance (increase in P_{EB}) as the rms value of the total correlation plus channel noise decreases. This is probably due to a change in distribution of the total noise: for small SNR, the channel

*In our discussion of Figures 11 and 12, we saw that this was equivalent to about four equal-strength, independent, Rayleigh paths when the signals generate no correlation noise. But the number K = 10 is a more appropriate number in the present discussion.

noise dominates, and the total noise output is essentially Gaussian. For large SNR, correlation noise dominates, and this may well have somewhat non-Gaussian distribution tails, controlled by the statistics of the multipath, leading to a larger P_{EB} for a given rms value than would Gaussian tails.

The "bottoming out" phenomenon just described shows up uniformly for all M'ary PDI receivers in Areas A and D at the predicted values of SNR. Since the argument that led to the prediction of the phenomenon did not refer to the type of receiver or the Area, one would expect it to hold for all Areas and receivers and for all M, even M = 2. (In the M = 2 case we have considered, DPSK is used, which involves only one matched filter and therefore only auto-correlation but not cross-correlation noise.) Surprisingly, "bottoming out" is clearly present in only one other curve in Figures 11-24, that for 16'ary DRAKE in Area A. This discrepancy between prediction and results arises because the argument predicting domination by correlation noise is somewhat oversimplified and because there are artifacts of the receiver simulations used that modify our conclusions.

For example, the LP receiver nominally seeks the value of the largest peak in the output of each matched filter. The simulation routine was simplified by assuming that this peak occurs at the "correct" filter's output at the time instant when the largest path's response would peak without noise; as in the binary case [1], this is not a bad approximation for the "correct" filter. Unfortunately, for further simplicity in the simulation, the other M-1 outputs were also sampled at the same time, rather than at the various (different) instants where each filter's output peaks. Sampling all outputs at the instant that the correct filter's output peaks is the worst possible choice since, for the signals chosen, $\rho_{mn}(0) = 0$ in (53) for all $m \neq n$. This means that, when sampling all filter outputs at the time of the largest path (when the correct filter outputs its largest peak), the cross-correlation noise induced by this largest path in all M-1 "incorrect" filters is exactly zero. Since the largest path contributes a very large fraction of the cross-correlation noise (recall that the paths are presumed to be more or less of equal strength only on the average, and even this is an oversimplication since one path usually dominates all others even on the average), we have wrongly suppressed in our simulation a substantial part of the correlation noise.

Indeed, in Area D, where there may be only one path, we have suppressed virtually all correlation noise. In other words, instead of sampling the largest values of cross-correlation noises in all of the M-1 "incorrect" filters, we have in our simulation likely sampled the smallest! The "bottoming out" phenomenon is thus artificially suppressed for LP receivers, at least in the SNR range we have investigated, although a slight upward inflection indicates its presence in the Area A curves.

For DRAKE, we see that "bottoming out" occurs for Area A, but not for Area D. Recall that the DRAKE receiver has a tapped delay line at the output of each matched filter's envelope detector; a tap is turned on for each path whose strength lies above a threshold that is at roughly the rms channel noise level. Thus, in Area A there are many taps turned on, at each of which appears substantial correlation noise from paths with delays corresponding to other taps; "bottoming out" occurs as predicted in Area A DRAKE performance.

However, as in the LP receiver, because $\phi_{mn}(0) = 0$ in (53) for all $m \neq n$, a path inducing a multipath peak at a given tap at the "correct" filter's output will induce no cross-correlation noise at the same tap at other filters' outputs. Thus, in Area D, where there is typically only one dominant path and perhaps only not even one other path above threshold, there is considerably less cross-correlation noise than predicted by the simplified theory or perhaps none at all; "bottoming out" will occur at a very much greater SNR than encompassed in our graphs.

Again, in a RAKE receiver, since tap gains on the delay line are adjusted to be equal to the associated path strengths, the largest potential component of correlation noise -- that from the strongest path -- is given the largest weight; and, since $\phi_{mn}(0) = 0$ for all $m \neq n$, that component is zero. Again, our theoretical prediction is oversimplified. We see that only a slight upward inflection in the RAKE curves for Area A hints at the "bottoming out" phenomenon, and not even this hint appears in Area D.

In sum, we see that our theoretical analysis of "bottoming out" -- which is based on a stationarity assumption on the correlation noise -- is most accurate for PDI receivers, which view large numbers of samples of correlation noise. The prediction is less and less accurate for DRAKE, RAKE and LP receivers, which view successively fewer samples (and in the LP case, a decidedly atypical sample) of the noise.

V. Discussion

The main results of our simulation experiments are twofold:

- Binary DPSK with RAKE or DRAKE performs as well as or better than any of the M'ary systems we have considered, even though binary transmission involves substantial intersymbol interference in the high-rate case.
- PDI reception, which offered hope for a simplified M'ary receiver in which intersymbol interference could be avoided, is distinctly not a viable choice.

We are thus left to a choice between binary and M'ary RAKE (or perhaps DRAKE, although DRAKE is more susceptible to correlation noise.)*

The choice between binary and M'ary RAKE is a subtle and complex one, involving the following issues.

(1) As M increases, the number of matched filters and RAKE algorithm circuits** in Figure 10 increases, but the channel estimator, which provides path strength/delay estimates to the algorithm circuits, becomes simpler. On the other hand, in the binary case there is only one matched filter and algorithm circuit, but the estimator structure becomes more complex [2]. There is thus a system complexity trade-off between binary and M'ary.

(2) Even in the M'ary case, there is a choice between standard coding (say 16'ary) and commutation coding (say (4+1)'ary). Here, choice of commutation coding reduces complexity but involves a power penalty.

(3) M'ary signalling involves a $\log_2 M$-to-1 reduction of bandwidth for the same processing gain TW, or a $\log_2 M$-to-1 increase in processing gain for the same TW product.

* LP reception might also be a viable choice in the M'ary case. However, because of the unfortunate assumption used in our LP simulation, discussed in Section 4.4, our simulation curves for LP reception are quite optimistic. Since even these curves are at least 5 dB worse in Area A than RAKE curves, we suspect that LP reception with M'ary signalling is also not a satisfactory choice.

** The algorithm circuits can be passive (i.e., tapped delay lines as in [1]) or active (real-time multiplication of the matched-filter output envelopes by the estimated multipath profile).

(4) While our simulation results have shown that 16'ary and binary RAKE systems perform more or less equivalently, one suspects that 16'ary systems are more susceptible to correlation noise from other systems operating in the same band. This conjecture is not a sure one; to the extent that inter-system correlation noise looks exactly like channel noise at all matched-filter outputs, it simply reduces the effective $\bar{\mathcal{E}}_{LOS}/N_0$ in any one system by an amount that is independent of M. To the extent that signal structures and channel fading statistics make inter-system noise look different from channel noise, it may be that systems with different M are differently affected.

Issue (1) is the subject of a concurrent study. Issues (2) and (3) are a matter for the system designer's judgement. Issue (4) has been proposed for further study.

Appendix: Calculations of Prob(no decoding correction/E → E)

There are six possible transmitted signal pairs, AB, AC, BA, BC, CA, CB, each having probability $\frac{1}{6}$ of occurring. The possible double-error patterns are:

Transmitted pair	Binary or ternary decisions	Only ternary decisions
AB	$\underline{\mathrm{BA}}$, $\underline{\overline{\mathrm{CA}}}$, $\underline{\overline{\mathrm{BC}}}$	CC
AC	$\overline{\mathrm{BA}}$, $\underline{\mathrm{CA}}$, $\underline{\overline{\mathrm{CB}}}$	$\underline{\mathrm{BB}}$(case 3)
BA	$\underline{\mathrm{AB}}$, $\overline{\mathrm{AC}}$, $\overline{\mathrm{CB}}$	CC
BC	$\overline{\mathrm{AB}}$, $\overline{\mathrm{CA}}$, $\underline{\mathrm{CB}}$	AA
CA	$\underline{\overline{\mathrm{AB}}}$, AC, $\overline{\mathrm{BC}}$	BB
CB	$\underline{\overline{\mathrm{AC}}}$, $\overline{\mathrm{BA}}$, $\underline{\mathrm{BC}}$	AA

In this table, the underlined errors lead to a corrected decoding using code #1 of Figure 8; the overlined errors lead to a corrected decoding using code #2; the AC → BB error is correctly decoded only when code #1 is used with ternary decisions and BB is decoded as a 1 (case 3, p. 24).

We now calculate Prob(no decoding correction/double error) for the three cases listed on page 24 (code #1) and the two cases listed on page 28 (code #2).

Case 1: Here, the probability of an error pattern depends on whether or not the first error precludes the actually transmitted second signal from being considered on the second decision. This is the case for the error pairs $\mathrm{AB} \rightarrow \underline{\mathrm{BA}}$, $\mathrm{AB} \rightarrow \overline{\mathrm{BC}}$, $\mathrm{AC} \rightarrow \underline{\mathrm{CA}}$, $\mathrm{AC} \rightarrow \underline{\overline{\mathrm{CB}}}$, $\mathrm{BA} \rightarrow \underline{\mathrm{AB}}$, $\mathrm{BA} \rightarrow \overline{\mathrm{AC}}$, $\mathrm{BC} \rightarrow \overline{\mathrm{CA}}$, $\mathrm{BC} \rightarrow \underline{\mathrm{CB}}$, $\mathrm{CA} \rightarrow \underline{\overline{\mathrm{AB}}}$, $\mathrm{CA} \rightarrow \mathrm{AC}$, $\mathrm{CB} \rightarrow \overline{\mathrm{BA}}$, $\mathrm{CB} \rightarrow \underline{\mathrm{BC}}$. The probability of each of these error pairs, given that there is a double error,is $\frac{1}{6}$ x (the prob. of the transmitted sequence) x $\frac{1}{2}$ (the prob. that the first error is what it is) x (the prob. that the second error is what it is) = $\frac{1}{24}$. If the first error does not preclude the actually transmitted second signal from consideration on the second signal ($\mathrm{AB} \rightarrow \underline{\overline{\mathrm{CA}}}$, $\mathrm{AC} \rightarrow \overline{\mathrm{BA}}$, $\mathrm{BA} \rightarrow \overline{\mathrm{CB}}$, $\mathrm{BC} \rightarrow \overline{\mathrm{AB}}$, $\mathrm{CA} \rightarrow \overline{\mathrm{BC}}$, $\mathrm{CB} \rightarrow \underline{\overline{\mathrm{AC}}}$), the probability of the double error is $\frac{1}{6}$ x $\frac{1}{2}$ x 1.* Thus, for code #1, the probability of a decoding

*If the correct signal is one of the two under consideration, there is only one way of making the second error.

correction, given that there is a double error is $\frac{6}{24} + \frac{2}{12} = \frac{5}{12}$, whence Prob(no dec. corr./E → E) = $\frac{7}{12}$. On the other hand, for code #2, Prob(no dec. corr./E → E) = $\frac{1}{4}$.

Case 2: Here, the probability of every error pair, given that there is a double error, is $\frac{1}{24}$. Thus, for code #1, Prob(no dec. corr./E → E) = $1 - \frac{8}{24} = \frac{2}{3}$; for code #2, Prob(no dec. corr/E → E) = $1 - \frac{12}{24} = \frac{1}{2}$.

Case 3: For code #1, there is one more way of correcting an error (AC→BB), so Prob(no dec. corr./E → E) = $1 - \frac{9}{24} = \frac{15}{24}$.

REFERENCES

[1] G. L. Turin, "Antimultipath Techniques for Packet Radio Systems," Packet Radio Technical Note #262, SRI International, Menlo Park, CA; November 22, 1978. See also, G. L. Turin, "Introduction to Spread-Spectrum Anti-Multipath Techniques and Their Application to Urban Digital Radio," Proc. IEEE, vol. 68, pp. 328-353; March 1980.

[2] G. L. Turin, "Antimultipath Techniques for Packet Radio, II: Summary of the Work of M. A. Kamil and L.-F. Wei," Packet Radio Technical Note #282, SRI International, Menlo Park, CA; December 17, 1979.

[3] J. Leung, "Throughput Improvement of Code-Division Multiple Access (CDMA) Packet Radio Channels...," Packet Radio Technical Note #227, SRI International, Menlo Park, CA; June 1977.

[4] G. L. Turin, "The Asymptotic Behavior of Ideal M'ary Systems," Proc. IRE, vol. 47, pp. 93-94, January 1959.

[5] P. M. Hahn, "Theoretical Diversity Improvement in Multiple Frequency Shift Keying," IRE Trans. on Communication Systems, vol. CS-10, pp. 117-184; June 1962.

[6] A. J. Viterbi, "Phase-Coherent Communication over Continuous Gaussian Channel," in "Digital Communications," S. W. Golomb, ed., Prentice-Hall, 1964; p. 128.

[7] J. Leung, "Megabit Packet Radio Receiver Design," SRI International, internal note; Sept. 1978.

[8] G. L. Turin, "Antimultipath Techniques in Packet Radio," Interim Report #2 to SRI International; February 13, 1977.

[9] R. Gold, "Optimal Binary Sequences for Spread-Spectrum Multiplexing," IEEE Trans. on Information Theory, vol. IT-13, pp. 619-621, Oct. 1967.